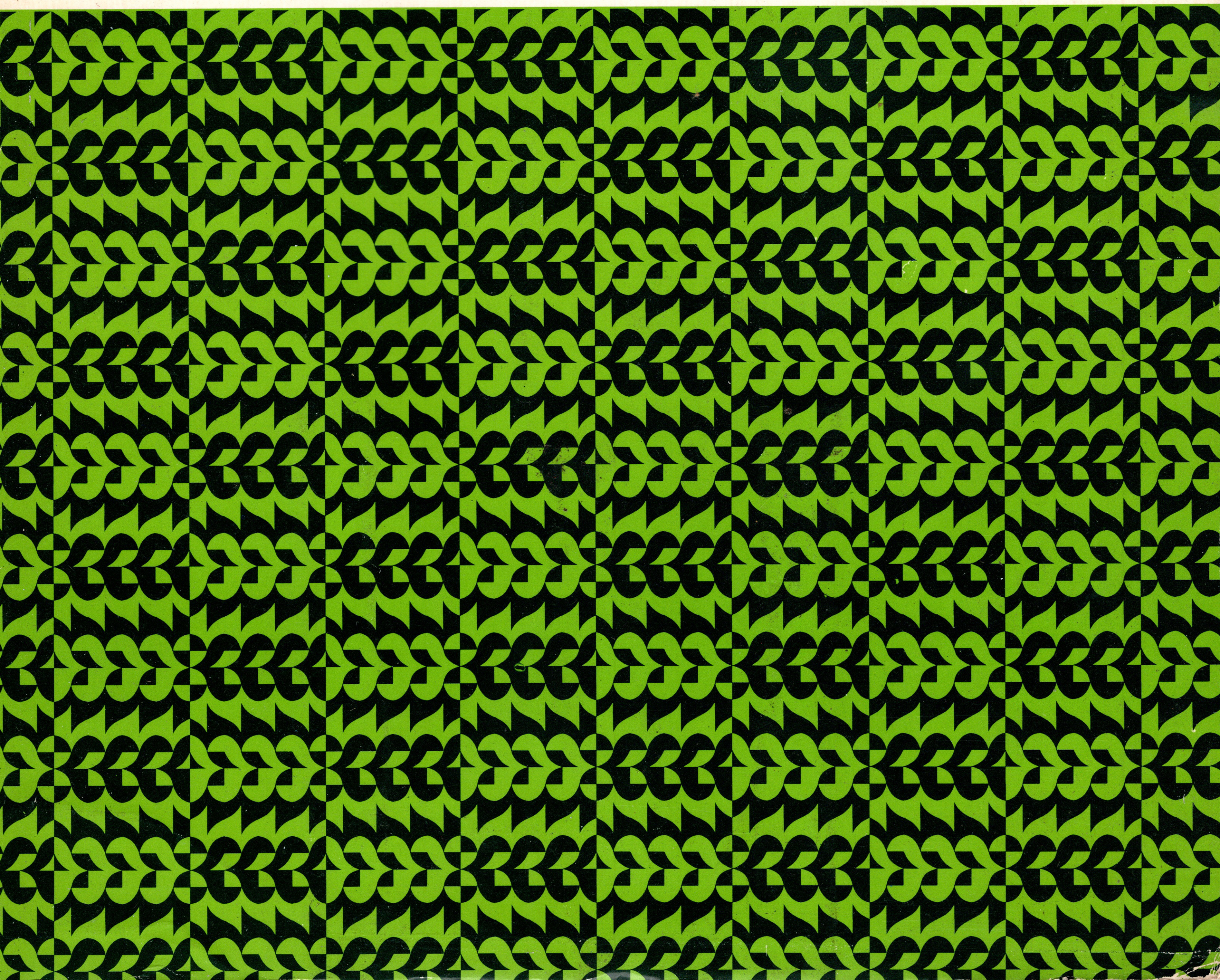
Computer Composition

Using PAGE—1

John Pierson



Computer Composition

Using PAGE–1

Computer Composition

Using PAGE–1

John Pierson
Princeton, N.J.

WILEY–INTERSCIENCE
A Division of John Wiley & Sons, Inc.
New York • London • Sydney • Toronto

Copyright © 1972, by John Wiley & Sons, Inc.

All rights reserved. This book or any part thereof must not be reproduced by any means, nor transmitted, nor translated into a machine language without the written permission of the publisher. However, any who may find the format statements useful are invited to use them freely in the machine readable form. They have been debugged carefully against most foreseeable typographic circumstances that they are likely to encounter in typographic composition. No guarantee of their workability in particular circumstances is extended or intended by this invitation.

Library of Congress Catalog Card Number: 74–179496
ISBN 0 471 68955 6
Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To Lissa and Tricia

PREFACE

This book is written as an introduction to composition programming; that is, the expression of book design specifications in the form of PAGE–1 procedures or "programs". People who practice what has come to be known as "formatting"—those who write procedures to direct the PAGE–1 processor in the composition of lines and pages of text for subsequent typesetting on an RCA VIDEOCOMP* Electronic Photocomposer—come generally from two groups. Some are computer programmers. Others come with one or several graphic arts backgrounds: editors and copy readers from the publishing houses and keyboard operators and make-up people from the typesetting houses. They are entering the third era of typesetting and publishing together; each with little or no knowledge in the other's area of experience.

This book presents a way of looking at the printed page and then describes the process by which it comes into being in the language of PAGE–1. The intent is to teach a kind of programming to people whose prime interest is in the graphic arts and at the same time to ease the entry of the programmer into the strange and wonderous world of typographic composition.

The PAGE–1 system of programs, more properly known as the Graphic Programming System, was produced by the Programming Development Group at Graphic Systems Division of RCA during 1966 and 1967. (The language of PAGE–1 owes much to Dr. M. P. Barnett who was at G.S.D. during 1965 and 1966.) The system was released to its first commercial user in the autumn of 1967 and was used in the composition of *The Long Short Cut* by Andrew Garve, published by Harper & Row, Publishers, Inc., in 1968. The book was produced by the Haddon Craftsmen, Inc., of Scranton, Pennsylvania; composition (using format statements written by this author) and typesetting (on a 70|820 VideoComp) were done at Videographic Systems, Inc., Hauppauge, L.I., New York. This publishing first was reported in the *New York Times* of 10 April 1968.

Sample problems, or case studies, are essential to books of this sort and this book is no exception. Application of the language to composition problems is centered around case studies selected as typical book problems. The problems chosen are by no means merely introductory nor are they trivial.

---

*VideoComp® is a registered Trademark of the RCA Corporation.

The proof of the pudding is ultimately in the sample pages. The author is greatly indebted to the McGraw–Hill Book Company; Scott, Foresman and Company; and John Wiley & Sons, Inc. All three extended permission to use and reproduce style specifications; to type published material as sample manuscripts; and to keyboard, compose and typeset the sample material.

Strategy for the Americas, by George W. Reidy and published by McGraw–Hill in 1966, is used in a number of illustrations throughout the book. This book is the basis of the development, in Chapter 7, of procedures for single-column text with footnotes.

MAINTAINABILITY: A Major Element of System Effectiveness, by A. S. Goldman and T. B. Slattery and published by John Wiley & Sons in 1964, appears in the sample pages of Chapter 10. This book was also the model for the procedures for allocation of white space in single-column text (Chapter 8) and for the author's experiments in the setting of mathematics.

"A Horseman in the Sky" by Ambrose Bierce was found by the author in *ACCENT: U.S.A.,* published by Scott, Foresman and Company in 1965. This book is the model for the two-column text with footnotes and white space (Chapters 7 and 8) and also for the "hanging folios".

The illustrations in Chapter 2 are adapted for the VideoComp from Volume 3 of *A Typographic Quest,* copyright 1965 by Westvaco Corporation. The text in the leading exhibit was excerpted from *An Essay on Typography,* by Eric Gill, published by Faber & Faber. The author expresses his thanks to Westvaco for permission to adapt the material.

The cover design was generated using a font the author named *Thing One.* The idea for generating modular art on a VideoComp and the design of *Thing One* arose from the fertile imagination of Mr. Stanley Rice to whom the author is indebted for its use here.

The book couldn't have been without the wholehearted and enthusiastic support of countless people at the Graphic Systems Division of RCA. Among these, the author wishes especially to express thanks to Mr. David Meredith, Manager of Programming Development, for encouragement and for machine time to test the case study material and to compose and typeset the illustrations and the book. To Mr. Paul Justus, Manager, Composition, for countless hours of consultation and for "pushing" the author's work in the Graphic Systems Center before the author's return to GSD. To Mr. R. S. Eiferd, Manager, Font Development, for development of special fonts for the author's experiments in mathematics. And by no means least, to Mr. D. S. Sikora, Manager, 70|830 Development, for his patient council and encouragement and for nearly a year of courier duty between the GSD plant in Dayton, N.J., and the author's home in Moorestown, N.J.

Princeton, New Jersey
October, 1971

JOHN PIERSON

TABLE OF CONTENTS

Introduction 1

Part II

The size and style of type 6
Elementary manuscript coding 10
Further PAGE–1 codes 16

Part III

Pagination algorithms 28
Subsidiary format statements 36
Procedures to set and to position footnotes 41
Procedures to allocate and to position white space 52

Part IV

Generalized format statements 62
Sample pages 67

Appendix A A PAGE–1 dictionary

Appendix B A PAGE–1 format manual

Appendix C Type specimens

Appendix D A glossary of terms

Computer Composition

Using PAGE–1

1

INTRODUCTION

A NEW GENERATION of typesetting devices has gained wide acceptance in the graphic arts industry; they all have in common the fact that they generate characters on the face of a high-resolution cathode ray tube. Lines or pages of such generated material are recorded on photosensitive film or paper for subsequent printing by the photo-offset method. One such device is the RCA Video-Comp Electronic Composition System.

These "third generation" typesetting machines lend themselves quite naturally to operation by a computer; indeed, they demand it. Their speed and flexibility can be exploited in no other way. Concurrently with the development of the first VideoComp, the RCA Graphic Systems Division developed the Graphic Programming System, GPS, in order to provide the industry a total system capability rather than simply a typesetting device. The application of this programming system to typesetting problems is the subject of this book.

Book production

Production of a book is governed by a document known as a style specification, which delineates such things as the width and depth of the printed area, the typeface and size to be used for text, folios and other headings, and footnotes; it also specifies details of format and typography. The style specification is the typesetter's guide for *setting* the book in type.

A simplified summary of the style specification used by the McGraw–Hill Book Company to direct their typesetter for the typesetting of *Strategy for the Americas* is shown. The information in parentheses is not normally supplied in a non-computer typesetting environment. Space between footnotes and text and around side heads and secondary text is available to the typesetter's composing room people to make pages come out to the proper depth when they make up pages. The computer system, however, needs stated minimum spaces since the program will produce fully made-up pages.

STYLE
SPECIFICATION

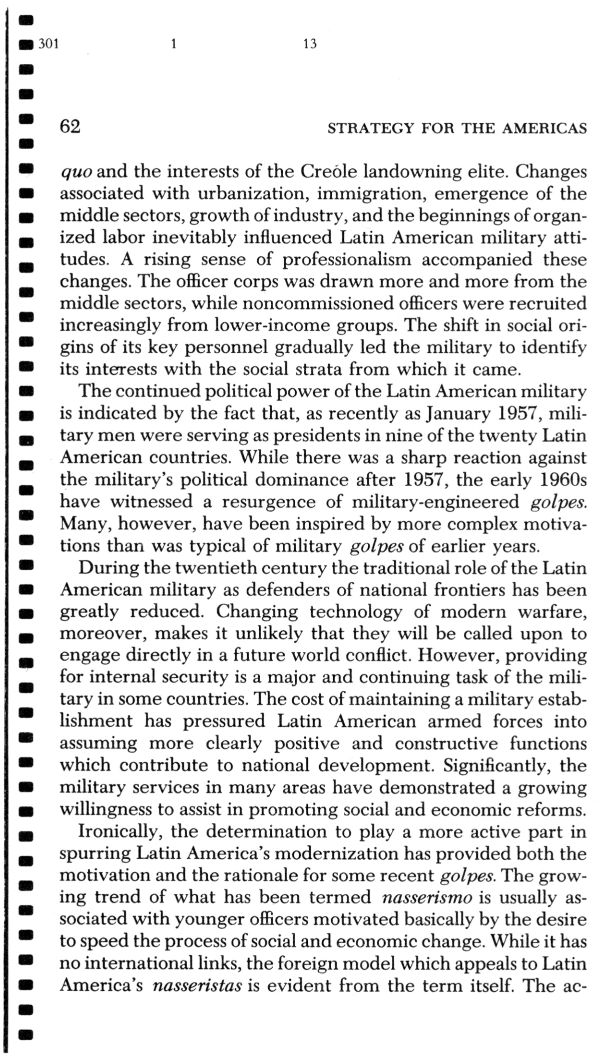
Type page	24 × 40 picas, 38 lines per page.
Text	10/12 Caledonia.
Extracts	9/11 Caledonia, indented 2 ems (4 points above and below).
Footnotes	8/10 Caledonia, cited by consecutive superscript Arabic numerals, first footnote in each chapter is numbered 1, minimum space 2 points above).
Folio	10 point Caledonia, flush outside, 14 points below.
Drop folio	Chapter opening, 10 point Caledonia, centered, 2 points above.
Running heads	10 point Caledonia, small caps, flush inside, left—book title, right—chapter title.
Chapter heads	
number	24 point Caslon Bold, 7 pica indent, 2½ pica sink from top of type page, 2 picas below.
title	18 point Caslon Bold, 7 pica indent, 2 picas above and below, capital and lower case, break into approximately equal lines.
Initials	None, chapter opening indented 7 picas.
Side heads	
first	10/12 Caledonia Bold, capital and lower case, flush left, 12 points above, 6 points below, turnovers flush left.
second	10/12 Caledonia Italic, capital and lower case, flush left, 10 points above, 2 points below, turnovers flush left.

Strategy for the Americas was published before the advent of the VideoComp; however, Chapter 4 has been typeset many times during PAGE–1 debugging and demonstration. Pages from Chapter 4, which were typeset on a 70|830 VideoComp, are reproduced here; these pages illustrate the things covered by the style specification. As these "sample pages" are discussed, a few definitions and units will be introduced.

The type page described by the style specification is the area on which characters are written and includes a running head and folio, or page number, at the top separated from a block of text by 14 points. The nominal text block; that is, one without extract text, side heads, or footnotes, contains 38 lines of text. The running head, folio, and primary text are all set in 10 point Caledonia. The primary text is *leaded* two points; that is, two points of additional space are provided between lines to improve readability.

In typography, type sizes are measured in points and distances in points and picas. In the United States, the point is approximately equal to 1/72nd of an inch. Twelve points are equal to one pica—approximately six picas to the inch. The size of type characters, or types, is measured from the top of the ascending letters (eg., lower case h) to the bottom of the descending letters (lower case p).

TYPE
PAGE



301 1 13

62 STRATEGY FOR THE AMERICAS

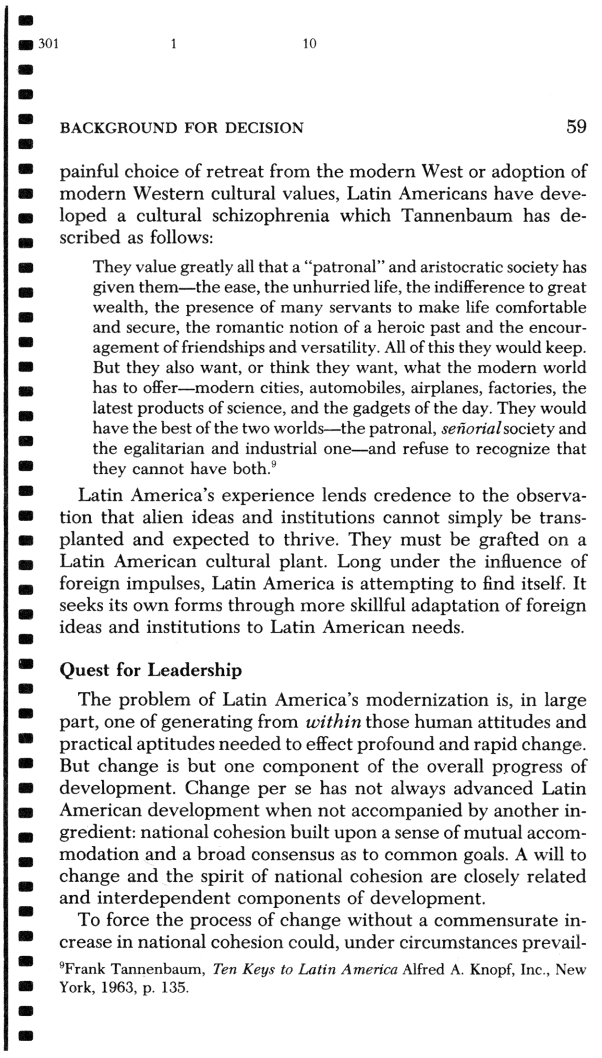
quo and the interests of the Creôle landowning elite. Changes associated with urbanization, immigration, emergence of the middle sectors, growth of industry, and the beginnings of organized labor inevitably influenced Latin American military attitudes. A rising sense of professionalism accompanied these changes. The officer corps was drawn more and more from the middle sectors, while noncommissioned officers were recruited increasingly from lower-income groups. The shift in social origins of its key personnel gradually led the military to identify its interests with the social strata from which it came.

The continued political power of the Latin American military is indicated by the fact that, as recently as January 1957, military men were serving as presidents in nine of the twenty Latin American countries. While there was a sharp reaction against the military's political dominance after 1957, the early 1960s have witnessed a resurgence of military-engineered *golpes.* Many, however, have been inspired by more complex motivations than was typical of military *golpes* of earlier years.

During the twentieth century the traditional role of the Latin American military as defenders of national frontiers has been greatly reduced. Changing technology of modern warfare, moreover, makes it unlikely that they will be called upon to engage directly in a future world conflict. However, providing for internal security is a major and continuing task of the military in some countries. The cost of maintaining a military establishment has pressured Latin American armed forces into assuming more clearly positive and constructive functions which contribute to national development. Significantly, the military services in many areas have demonstrated a growing willingness to assist in promoting social and economic reforms.

Ironically, the determination to play a more active part in spurring Latin America's modernization has provided both the motivation and the rationale for some recent *golpes.* The growing trend of what has been termed *nasserismo* is usually associated with younger officers motivated basically by the desire to speed the process of social and economic change. While it has no international links, the foreign model which appeals to Latin America's *nasseristas* is evident from the term itself. The ac-

EXTRACT TEXT,
SIDE HEAD, & FOOTNOTE



301 1 10

BACKGROUND FOR DECISION 59

painful choice of retreat from the modern West or adoption of modern Western cultural values, Latin Americans have developed a cultural schizophrenia which Tannenbaum has described as follows:

> They value greatly all that a "patronal" and aristocratic society has given them—the ease, the unhurried life, the indifference to great wealth, the presence of many servants to make life comfortable and secure, the romantic notion of a heroic past and the encouragement of friendships and versatility. All of this they would keep. But they also want, or think they want, what the modern world has to offer—modern cities, automobiles, airplanes, factories, the latest products of science, and the gadgets of the day. They would have the best of the two worlds—the patronal, *señorial* society and the egalitarian and industrial one—and refuse to recognize that they cannot have both.[9]

Latin America's experience lends credence to the observation that alien ideas and institutions cannot simply be transplanted and expected to thrive. They must be grafted on a Latin American cultural plant. Long under the influence of foreign impulses, Latin America is attempting to find itself. It seeks its own forms through more skillful adaptation of foreign ideas and institutions to Latin American needs.

Quest for Leadership

The problem of Latin America's modernization is, in large part, one of generating from *within* those human attitudes and practical aptitudes needed to effect profound and rapid change. But change is but one component of the overall progress of development. Change per se has not always advanced Latin American development when not accompanied by another ingredient: national cohesion built upon a sense of mutual accommodation and a broad consensus as to common goals. A will to change and the spirit of national cohesion are closely related and interdependent components of development.

To force the process of change without a commensurate increase in national cohesion could, under circumstances prevail-

[9] Frank Tannenbaum, *Ten Keys to Latin America* Alfred A. Knopf, Inc., New York, 1963, p. 135.

Extract text, also called secondary text, is used when quoting another source or when, for some other reason, it is desirable to set material apart from primary text. Extract text is generally set in a smaller point size in the same typeface and it is sometimes indented as well. Sub heads are usually, though not always, set in a different typeface and they are given added space above and below to make them stand out on the printed page.

Footnotes are commonly set at the bottom of the page on which they are cited and in a smaller point size in the same typeface used for primary text. The footnote is usually cited in text by a superior arabic numeral and the footnote itself is introduced by the same number which may be a superior number or may be in the same point size used for the footnote text. The first line of a footnote may or may not be indented.

Some features of PAGE–1

The PAGE–1 language comprises about ninety function codes. Each is known by a unique two-letter name; the names were chosen to relate as much as possible to graphic arts terms to simplify use of the language by graphic arts oriented people and they were held to two characters to minimize the number of keystrokes required in any given job.

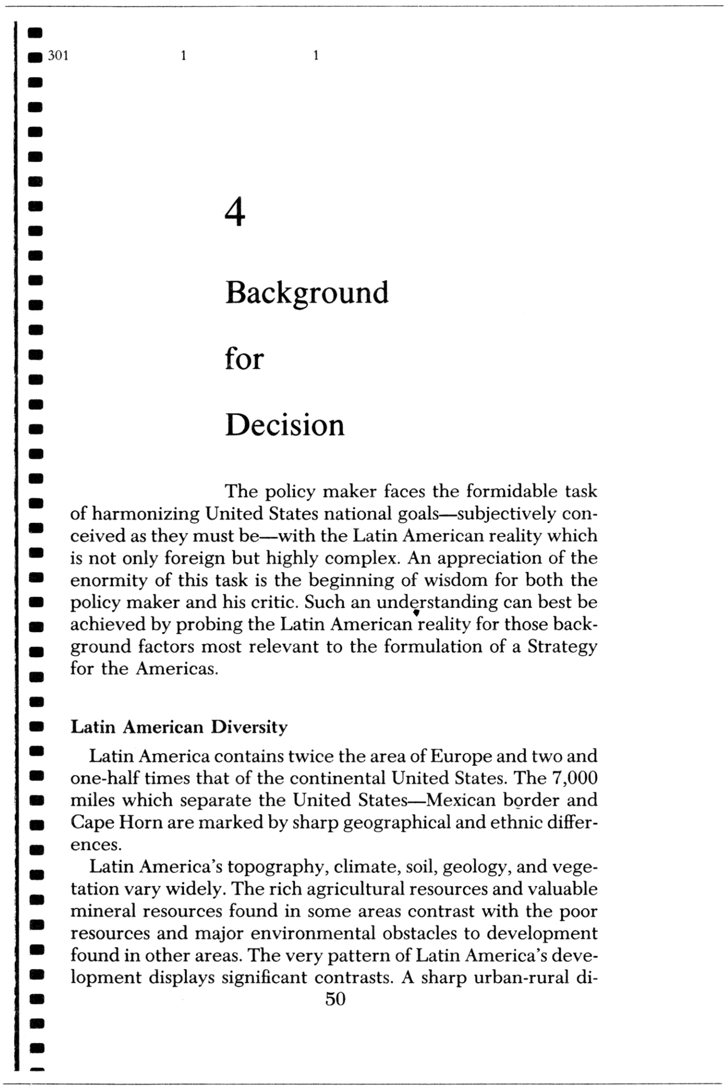
A few similarities and dissimilarities with the more common computer languages bear mention. Perhaps the most striking dissimilarity is the concept of what constitutes a program. In the usual sense, the program and the data are physically separate and manipulation of the data remains under direct control of the program. In the case of PAGE–1, the *data* (ie., the text to be typeset) and the *program* (PAGE–1 control statements or procedures) are mixed in one continuous input stream. This input stream is analyzed and processed by an interpretive processor. The processor operates on the text according to well defined rules and transfers control to the user *program* when PAGE–1 control codes are encountered in the input stream. These control codes, or program statements, enable the user to modify the rules for setting text, thereby altering succeeding composition functions.

The user program loses control during the text processing operation and regains control at those points where program statements occur in the input stream. An important and powerful extension of this process is a PAGE–1 feature which enables the user program to regain control when textual material encounters either the right hand edge or the bottom of the setting area. The user's *bottom boundary routine* is particularly useful, since it is in this routine that the pagination procedure may be specified and modified.

Two distinct delimiters are used to demark control from data or manuscript—an opening or left delimiter and a closing or right delimiter. They are represented in manuscript by left and right square brackets. The same delimiters are used to accomplish nesting within control strings. Manuscript data then is nest level zero and nest level one is control. In the scheme of things, odd numbered nest levels are control and even numbered nest levels contain text to be typeset.

There is a number of similarities between PAGE–1

CHAPTER
OPENING



301 1 1

4

Background

for

Decision

The policy maker faces the formidable task of harmonizing United States national goals—subjectively conceived as they must be—with the Latin American reality which is not only foreign but highly complex. An appreciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

Latin American Diversity

Latin America contains twice the area of Europe and two and one-half times that of the continental United States. The 7,000 miles which separate the United States—Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

Latin America's topography, climate, soil, geology, and vegetation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's development displays significant contrasts. A sharp urban-rural di-

50

and other computer languages. PAGE–1 has the capability to store numbers. The mechanism is called *general variable;* there are 200 of them numbered 1 through 200. They simply provide a way to store, retrieve, and manipulate numbers. Two dedicated variables are short handed **pn** and **fn** for page number counter and footnote (citation) counter respectively.

It is also possible to define, store, and retrieve strings of characters. The definition and storage of strings is effected through two function codes to mark the beginning and end of the desired string. These codes allow for the material to be typeset as it occurs in the input stream and to be saved for subsequent use as may be required by the style specification. The string may comprise control as well as text. Another code is provided to *execute* the *saved string.* An example of the use of this facility in typesetting occurs in dictionaries and directories in which the first and last entries on a page are often displayed prominently at the top of the page.

PAGE–1 has the capability to do eight elementary arithmetic operations. These are sufficient to accommodate the requirements of page composition. The language also has the facility to execute controls based on the satisfaction of one of six stated conditions. The ability to test current conditions and take alternative actions based on the results is fundamental to the power of any pre-defined program. It is essential to PAGE–1 programs as well. Conditional statements may be made within boundary procedures, saved strings, or other conditional statements. Up to twenty nest levels are permitted in the current system.

PAGE–1 has a *macro-like* capability in that long strings of controls which will be used repetitively throughout the job can be given a label and called. This is not a macro generation capability to be sure, but it provides for the insertion of a defined string of controls at a specific position in the text stream by putting the label at the desired place. Two such capabilities are provided. One, which is global, stored in the system, and available to any job, is called a *Format Statement.* The other, which is explicitly defined and stated at the beginning of the data for a given job, is called a *Synonym.*

When procedures, or programs, have been written to satisfy the requirements of the style specification, they may be entered into the system as format statements or with the data for a given job as synonyms. The author's manuscript is "marked up" using the appropriate tags and in some cases explicit PAGE–1 codes. The keyboard operator types the manuscript, including the controls, into paper tape, magnetic tape, or punched cards. When the manuscript has been entered into the system, composition may begin.

Alterations

PAGE–1 includes a number of control codes to provide for changing composed data. Data to be corrected, or otherwise altered, is referenced by word space numbers within composed lines. Lines are referenced by numbers which are generated during the composition process and which may be found on output proof copy or on a special printer listing of the composed file.

The alteration controls are described in Appendix A; but they will not be discussed in detail in this presentation of the language. In general, PAGE–1 users have found a separate text editing program more convenient to use. This fact was acknowledged in a successor system, PAGE–2; the alteration commands were eliminated altogether.

PART II

THE PAGE–1 VOCABULARY

Part II introduces the PAGE–1 commands or control words within the general framework of their functions or uses in the practice of typography. The presentation does not presume to treat graphic design nor the history of typography. There are many fine references available to those who enter this field without knowledge in the graphic arts. Generally the composition programmer is called upon to "code" the specification of a designer; in effect to translate the designer's style specification into PAGE–1 code. On the other hand, those who "grew up" in the graphic arts industry will find a language in which the terminology is familiar and which facilitates delineation of the functions or tasks of composition and page make-up with which they are already familiar. In brief, a PAGE–1 procedure simply describes the manual typesetting and composing room tasks which would be required to produce the same final typeset product.

Chapter Two introduces the basic building blocks of typography—the characters with which text is typeset—and the PAGE–1 codes which describe them.

Chapter Three considers the minimum manuscript mark-up required to generate the traditional galley and then to break-up that galley into page proofs. This exercise leads into Chapter Four and provides a foundation for the developments of Part III.

Chapter Four is a presentation of the remainder of the PAGE–1 commands. The primitive commands are presented in sets and in the general framework of the functions they perform in the composition of lines, the generation of pages, and the creation of special typographic format requirements.

2

THE SIZE AND STYLE OF TYPE

THE SIZE AND STYLE OF TYPE in which a manuscript is to be composed, and subsequently typeset, are governed by four PAGE-1 codes; three other codes specify the width of the interword spaces provided by the processor in building lines of type. Additionally, the VideoComp has the capability to generate a pseudoitalic display of characters which were digitized and are stored in erect or Roman form. Characters displayed in this way are said to be *obliqued.* Two codes specify use of and recall from this mode of machine operation.

The sizes of type

The *point* is the unit of type measurement; in the United States it is approximately 1/72nd of an inch. When a type size is stated, the measurement refers (historically) to the full, vertical dimension of the shank of metal on which the character was cast. In current phototypesetting practice, the size may be considered to be the depth of a "box" or a "frame" with constant depth and variable width, within which a character sits. The depth extends from the top of the ascending letters (b, d, f, h, k, l) to the bottom of the descending letters (g, j, p, q, y). The ascenders and descenders for a given font may or may not extend to the upper and lower limits of the point size frame.

The characters which have neither ascenders nor descenders (eg., a, o, x) occupy the central portion of the frame. They rest, or sit, upon an imaginary line tangent to their bases and extend upward to an imaginary line tangent to their tops (except for **t** which breaks the upper line). This central portion is called the x-height of the character set and it varies significantly from one typeface to another.

SIZE IN POINTS	
6	abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwx
7	abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmn
8	abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdef
9	abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyza
10	abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuv
12	abcdefghijklmnopqrstuvwxyzabcdefghijklmn
18	abcdefghijklmnopqrstuvwxyz
24	abcdefghijklmnopqrstu
36	abcdefghijklmn
48	abcdefghijk
72	abcdefg
96	abcde

The size of type, *point size,* in which a manuscript is to be set is specified by the PAGE–1 code **ps**. The point size code requires a modifier, or *parameter,* to complete the specification of size. The PAGE–1 processor interprets the parameter in points. That is, **[ps,10]** specifies that the input material is to be set in ten point type.

VideoFonts are digitized in several size ranges—five for the 70|800 family—to cover the range of point sizes which the VideoComp is capable of setting. Each of the five size ranges is a separate font and the point size specified for use at any time must be consistent with the font in use. The size ranges I through V cover sizes 4 through 8, 8 through 16, 16 through 32, 32 through 64, and 64 through 96 points respectively.

Space between the lines

Additional space between lines of text is considered desirable and in some cases essential to the setting of easily legible text. Again using the "metal" analogy, if succeeding lines of type are laid together, the descenders of one line are very close to and may actually touch the ascenders of the succeeding line.

A natural channel of white space will exist between succeeding lines of type because most of the characters occupy only the central portion of the type frame. This natural channel becomes narrower as the x-height of the typeface becomes larger. The readability of such fonts is greatly improved by extra space between the lines.

LEADING

1-point lead
1½-point lead
2-point lead
3-point lead

10 pt. Video Gothic set solid (no leading)	It may be said of all printers that their job is to reproduce on paper the exact face of the letters which they have set into pages. This face is of a definite, con-
10/11 Video Gothic leaded 1 point	It may be said of all printers that their job is to reproduce on paper the exact face of the letters which they have set into pages. This face is of a definite, constant, and meas-
10/12 Video Gothic leaded 2 points	It may be said of all printers that their job is to reproduce on paper the exact face of the letters which they have set into pages. This face is of a definite, constant, and meas-
10/13 Video Gothic leaded 3 points	It may be said of all printers that their job is to reproduce on paper the exact face of the letters which they have set into pages. This face is of a definite, constant, and meas-

STYLES OF TYPE

Video Technica Medium	Electronic typography
Video Roma	Electronic typography
Video Times Roman	Electronic typography
Video Garamond	Electronic typography
Video Spectra	Electronic typography
Video News Gothic	Electronic typography
Video Baskerville	Electronic typography
Video Stymie Medium	Electronic typography
Video Gothic	Electronic typography
Video Janson	Electronic typography
Video Century Expanded	Electronic typography
Video Century Schoolbook	Electronic typography
Video Gael	Electronic typography
Video Clarendon	Electronic typograph
Video Modern	Electronic typogr

In the metal analogy, such additional space is provided by adding thin strips of metal between the lines of type. Such strips are available in thicknesses of 1, 1½, 2, and 3 points. The term *leading* derives from this insertion of additional metal between lines. Metal types may be cast on a body which is deeper than the point size to accomplish the same effect. For example, if the point size is eight and two points of leading are desired, the types may be cast on a ten point "body". The *body lead* is said to be ten points; the resulting types are eight point on a ten point body or 8/10.

In VideoComp usage it is convenient to think in terms of the metal analogy, yet body lead is in fact one basis for a film advance. (There are others as we shall see.) Body lead is expressed using the code **bl** with a parameter. The code string **[ps,10;bl,12]** specifies ten point type on a twelve point body; that is, two points of interline lead. (Note here, use of the semi-colon to separate two primitive code statements within a control "string".)

The style of type

Estimates of the number of typefaces in existance today range to 6000 of which only a fraction are in general use. In metal type, a typeface is available in a number of *fonts* from 6 to 72 points. A font in this sense is a collection of characters: upper and lower case letters, digits, points, and other special characters in one size and style of type. The font may include small capitals and italic as well.

There are upwards of 400 fonts representing more than 100 typefaces presently available for the 70|800 family of VideoComps. In this context the word font has a different meaning. Type does not exist as a physical entity in the VideoComp electronic phototypesetter. Rather, characters are generated on the face of a high-resolution cathode ray tube. Characters are formed by an electron beam which makes short, parallel strokes so fine and spaced so closely together that they appear solid. The beam is blanked and unblanked during succeeding scans of the type frame occupied by any given character. The data which provide this information to the machine are in fact the digital font. The forms of individual characters change as the point size changes from minimum to the maximum size the machine is capable of setting. Five individual fonts, or size ranges, cover the total range of the machine. Each of these five size ranges is a separate font. When a font is loaded on the system, it may be given an identifying number in the range one through 255 inclusive; but the number of fonts on any one system is not likely to exceed fifty. The limiting factor is the disc storage required to accommodate the fonts stored.

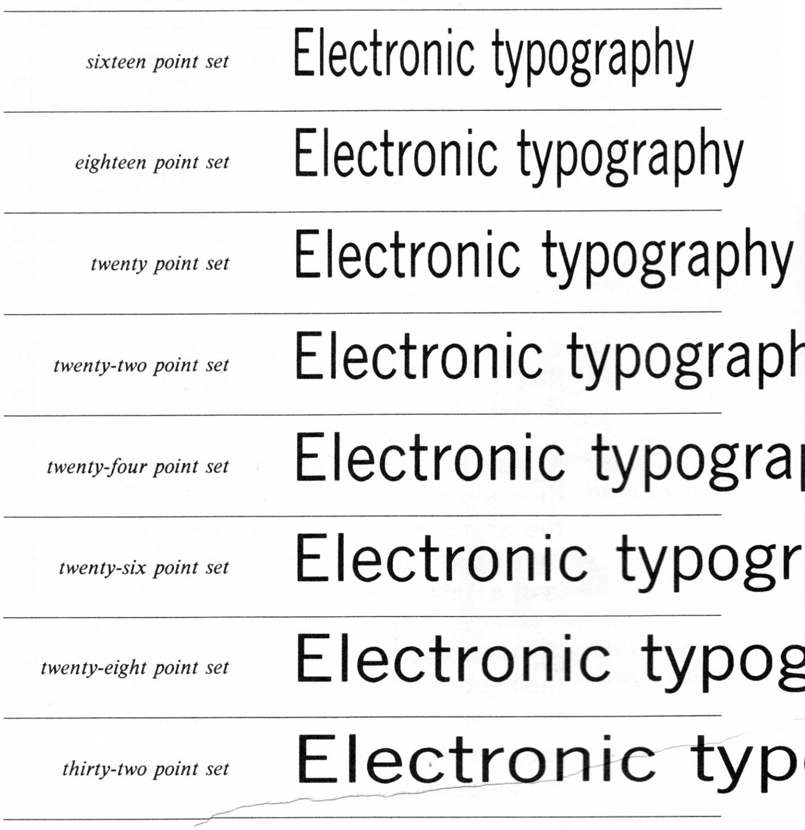
The typeface is specified using the code **tf** with a parameter which refers to the number assigned the font when it was loaded on the system. For example, **[tf,4]** in the system used to produce this book is Video Times Roman, Size Range II.

The set of the type

The *set* of a given character is its width. The letter **i**, for example, has a much narrower set than has the letter **w**. Until the advent of electronic photocomposing machines, the sets of the characters in a font, once established by the type designer, were fixed. A narrow type, of course, provides economy of space and a wider type leads to rounder more open characters. In general, the greater the x-height, the wider the set of the characters in the font.

The VideoComp has the capability to expand or condense the set width of characters as they are typeset. The width and depth of the characters in any VideoFont are defined such that they may be adjusted

SET WIDTH



for each of the valid point sizes in which the font can be set. Each time the point size is stated; for example, **[ps,8]**, the set width will be set to the same value. In general, then, the user need not concern himself with set width unless he wants it to be some value other than point size. If he does, set width must be reset to the desired value each time that point size is stated.

Set width is specified using the code **sw** with a parameter. The code string **[ps,8;sw,10]** will yield eight point characters which have been expanded to the widths specified for ten point type in the same font.

The italic characters

The compressed, sloped letter which we know today as *italic* was conceived and introduced near the end of the fifteenth century by Aldus Manutius. His desire was to create a type which would make possible the printing of smaller and therefore cheaper books. He patterned his new type after the informal handwriting of his time and called it *Chancery.* In Europe, the face was called *italic,* a designation which continues to the present day.

Though true italic counterparts are available for most of the VideoFonts, the VideoComp provides a machine capability to "slope" or "incline" characters which have been digitized erect. This feature is called *oblique* and its use is specified by the code **[ob]**. The angle introduced by the VideoComp is a factory adjustment which is normally about seventeen degrees. The code **[ro]** causes recall to normal, or erect, after the oblique mode has been in force.

True italic characters are, in fact, stored as erect characters and are typeset in the oblique mode. The **[ob]** code is transmitted automatically by the processor when a true italic font is specified for use.

Space in the line

The spaces which separate words in a line of type are in fact characters too though one usually doesn't think of them as such. The width of the "optimum" space is a font dependent consideration—fonts with a narrow set width require a narrower wordspace than do fonts with a wide set width. The width of the interword space is a compromise between a space narrow enough to encourage the eye in its scan of a line and yet wide enough that word patterns are distinct. The problem is compounded by the requirements of line justification in which all lines are forced to be the same length by compression and expansion of the interword space.

Three values for the interword space are specified for and supplied with each VideoFont. When the font is loaded on the system, these recommended (or other) values are stored with the font data. When the font is specified for use (that is, for example, **[tf,4]**), these stored, standard values are assigned to three codes called the *minimum interword space,* the *basic interword space,* and the *maximum interword space.* The user may use these three codes to change the values of the interword spaces.

The interword spaces are specified in a relative unit which is equal to the number of strokes with which the VideoComp writes a full em. This unit varies with the size range of the font. For the 70|800 the values are 100, 200, 400, 800, and 1600 units per em for size ranges one through five inclusive. In the system used in the composition of this book, typeface four is Video Times Roman, Size Range II. In this system, then, the code string **[tf,4;mi,50;bs,100;mx,150]** will call typeface four and then change the interword spaces to ¼, ½, and ¾ em respectively. The next time that typeface four is called, the standard values will again be assigned to the three interword space codes; if the user wants the altered values, he must change them again.

OBLIQUE
AND TRUE ITALIC

Video Roma	Electronic typography
Video Roma Oblique	*Electronic typography*
Video Roma Italic	*Electronic typography*
Video Times Roman	Electronic typography
Video Times Roman Oblique	*Electronic typography*
Video Times Roman Italic	*Electronic typography*
Video Gael	Electronic typography
Video Gael Oblique	*Electronic typography*
Video Gael Italic	*Electronic typography*

3

ELEMENTARY MANUSCRIPT CODING

ESSENTIAL SYSTEM VARIABLES are given initial or default values which will be in force when a job is started unless explicit instructions are placed in input with or prior to manuscript material to be composed. A *keyboard identifier* and a *start take* statement are all that must precede manuscript data in order to get typeset material from the PAGE–1 VideoComp system. The keyboard identifier is a single character which identifies the user's keyboard (and a translation table in the processor); the keyboard identifier must be the first character on an input tape. Following immediately is the start take statement which will be considered in detail subsequently; for now its form will be assumed.

If text to be typeset is entered into input following the keyboard indetifier and the start take statement, and with no additional controls, the typeset output produced by the VideoComp will have the following characteristics:

- The output will be *justified* (that is, the left and right margins will be straight) and hyphenation will be used in the process of justification.
- The output will be set in eight point type using typeface four, which in the system used to typeset this book and the samples is Video Times Roman, leaded two points, and the lines will be 25 picas (approximately four and one-fourth inches) long,
- If the input manuscript is of sufficient length, "page breaks will be issued automatically each time the text column becomes 100 picas (approximately 16 inches long.

This sort of output has little practical utility other than possibly for use as proof copy. Assuming that this set of default conditions is adopted however, the PAGE–1 command to start a new paragraph is likely the only other code necessary to get "useful" output. The *new paragraph* code is **[pr]**. The new paragraph command directs the processor to terminate the line being composed and start a new line spaced downward by the amount of the point size and leading in force and spaced from the left margin by the standard paragraph indention. The default value for the paragraph indention is one em. An example of several paragraphs under default conditions is shown in manuscript and final form in Figure 3.1.

If the output is desired in a different typeface and size and with a different *measure,* or line length, the codes to accomplish this can simply be written on the manuscript following the start take statement and preceding the manuscript. For example, if the output were desired in ten point Video Gael leaded two points and to a 25 pica measure, the code string **[ps,10;bl,12;tf,18;rb,300]** would be written on the manuscript. The code **ps** together with its parameter, **10**, specifies that the manuscript is to be composed in ten point type. The code **bl** with the parameter **12**, specifies a body lead or line advance of twelve points. This combination yields an additional two points between lines. Video Gael is specified by its number in the font file by assigning the parameter **18** to the code **tf**. Note, once more, that the typeface/number correspondence is characteristic of the user's system. In the system used to compose this book, Video Gael, Size Range II, is typeface eighteen. Finally, the line length is specified to be 300 points (25 picas) by a right boundary of 300 points together with a left boundary (default value) of zero. See Figure 3.2 for manuscript and output.

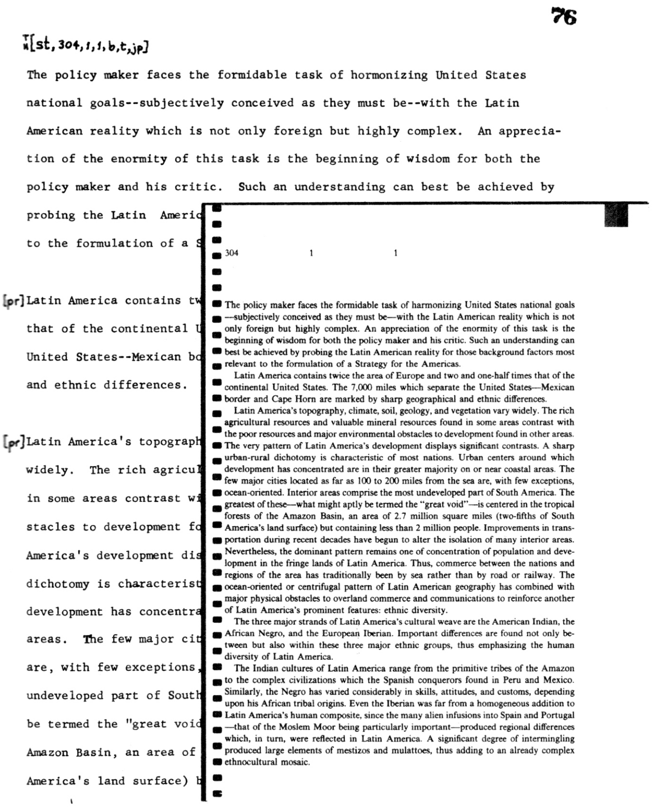
The remainder of this chapter is a description of one method of generating page proofs. This approach to composition with PAGE–1 is not recommended; rather, it is an introduction to the use and operation of some of the codes in the language.

Galley proof

A chapter number and title may be added at this point to produce something resembling the traditional galley. Assume the chapter number is to be set in 24 point and the title is to be set in 18 point Video Times Roman. The codes to specify typeface and size and the additional manuscript material may be added just preceding the code string of Figure 3.2. The chapter number, title, and opening line are generally spaced for esthetic effect; assume that the desire is for two picas between the chapter number and title and for two picas between title and opening with 1½ picas between lines of the title.

The codes to prepare for setting the chapter number are **[ps,24;bl,48;tf,166].** A body lead of 48 points will yield 24 points, or two picas, between the first and second lines; that is, between the chapter number and the first line of the title. Typeface **166,** in the system

Figure 3.1
Default conditions



76

```
[st,304,1,1,b,t,jp]
The policy maker faces the formidable task of hormonizing United States
national goals--subjectively conceived as they must be--with the Latin
American reality which is not only foreign but highly complex.  An apprecia-
tion of the enormity of this task is the beginning of wisdom for both the
policy maker and his critic.  Such an understanding can best be achieved by
probing the Latin  Americ
to the formulation of a S

[pr]Latin America contains tw
that of the continental U
United States--Mexican bo
and ethnic differences.

[pr]Latin America's topograph
widely.  The rich agricul
in some areas contrast wi
stacles to development fo
America's development dis
dichotomy is characterist
development has concentra
areas.  The few major cit
are, with few exceptions,
undeveloped part of South
be termed the "great void
Amazon Basin, an area of
America's land surface) b
```

304 1 1

The policy maker faces the formidable task of harmonizing United States national goals—subjectively conceived as they must be—with the Latin American reality which is not only foreign but highly complex. An appreciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

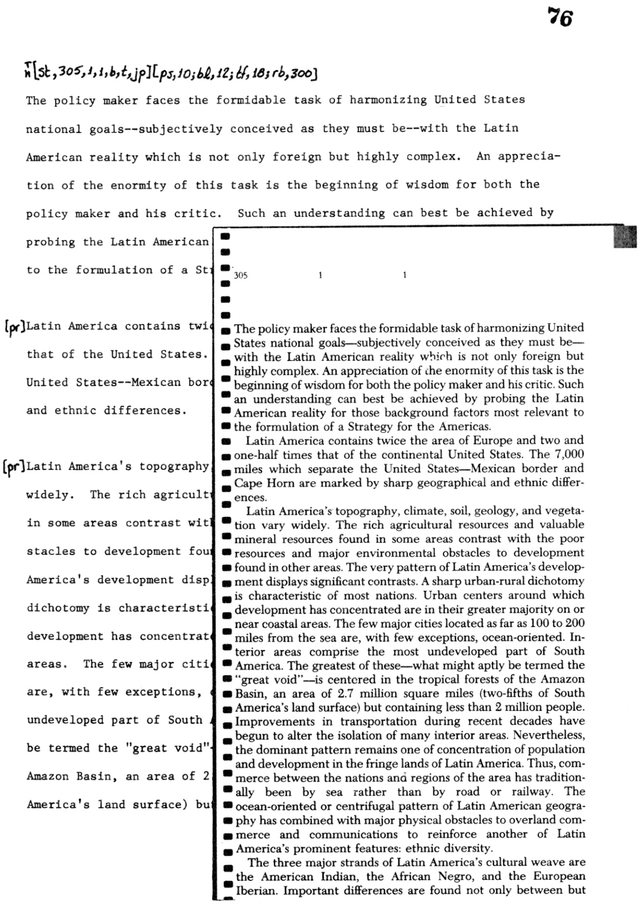
Latin America contains twice the area of Europe and two and one-half times that of the continental United States. The 7,000 miles which separate the United States—Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

Latin America's topography, climate, soil, geology, and vegetation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's development displays significant contrasts. A sharp urban-rural dichotomy is characteristic of most nations. Urban centers around which development has concentrated are in their greater majority on or near coastal areas. The few major cities located as far as 100 to 200 miles from the sea are, with few exceptions, ocean-oriented. Interior areas comprise the most undeveloped part of South America. The greatest of these—what might aptly be termed the "great void"—is centered in the tropical forests of the Amazon Basin, an area of 2.7 million square miles (two-fifths of South America's land surface) but containing less than 2 million people. Improvements in transportation during recent decades have begun to alter the isolation of many interior areas. Nevertheless, the dominant pattern remains one of concentration of population and development in the fringe lands of Latin America. Thus, commerce between the nations and regions of the area has traditionally been by sea rather than by road or railway. The ocean-oriented or centrifugal pattern of Latin American geography has combined with major physical obstacles to overland commerce and communications to reinforce another of Latin America's prominent features: ethnic diversity.

The three major strands of Latin America's cultural weave are the American Indian, the African Negro, and the European Iberian. Important differences are found not only between but also within these three major ethnic groups, thus emphasizing the human diversity of Latin America.

The Indian cultures of Latin America range from the primitive tribes of the Amazon to the complex civilizations which the Spanish conquerors found in Peru and Mexico. Similarly, the Negro has varied considerably in skills, attitudes, and customs, depending upon his African tribal origins. Even the Iberian was far from a homogeneous addition to Latin America's human composite, since the many alien infusions into Spain and Portugal—that of the Moslem Moor being particularly important—produced regional differences which, in turn, were reflected in Latin America. A significant degree of intermingling produced large elements of mestizos and mulattoes, thus adding to an already complex ethnocultural mosaic.

Figure 3.2
Simple Galley



76

```
[st,305,1,1,b,t,jp][ps,10;bl,12;tf,18;rb,300]
The policy maker faces the formidable task of harmonizing United States
national goals--subjectively conceived as they must be--with the Latin
American reality which is not only foreign but highly complex.  An apprecia-
tion of the enormity of this task is the beginning of wisdom for both the
policy maker and his critic.  Such an understanding can best be achieved by
probing the Latin American
to the formulation of a St

[pr]Latin America contains twi
that of the United States.
United States--Mexican bor
and ethnic differences.

[pr]Latin America's topography
widely.  The rich agricult
in some areas contrast wit
stacles to development fou
America's development disp
dichotomy is characteristi
development has concentrat
areas.  The few major citi
are, with few exceptions,
undeveloped part of South
be termed the "great void"
Amazon Basin, an area of 2
America's land surface) bu
```

305 1 1

The policy maker faces the formidable task of harmonizing United States national goals—subjectively conceived as they must be—with the Latin American reality which is not only foreign but highly complex. An appreciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

Latin America contains twice the area of Europe and two and one-half times that of the continental United States. The 7,000 miles which separate the United States—Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

Latin America's topography, climate, soil, geology, and vegetation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's development displays significant contrasts. A sharp urban-rural dichotomy is characteristic of most nations. Urban centers around which development has concentrated are in their greater majority on or near coastal areas. The few major cities located as far as 100 to 200 miles from the sea are, with few exceptions, ocean-oriented. Interior areas comprise the most undeveloped part of South America. The greatest of these—what might aptly be termed the "great void"—is centered in the tropical forests of the Amazon Basin, an area of 2.7 million square miles (two-fifths of South America's land surface) but containing less than 2 million people. Improvements in transportation during recent decades have begun to alter the isolation of many interior areas. Nevertheless, the dominant pattern remains one of concentration of population and development in the fringe lands of Latin America. Thus, commerce between the nations and regions of the area has traditionally been by sea rather than by road or railway. The ocean-oriented or centrifugal pattern of Latin American geography has combined with major physical obstacles to overland commerce and communications to reinforce another of Latin America's prominent features: ethnic diversity.

The three major strands of Latin America's cultural weave are the American Indian, the African Negro, and the European Iberian. Important differences are found not only between but used to generate these samples, is Video Times Roman, Size Range III. See Figure 3.3 for manuscript and typeset output.

After the chapter number, the codes **[nl;ps,18;bl,36]** prepare for chapter title. The *new line* code terminates the chapter number line and initiates a new line at the left margin and spaced downward by the leading in force. (A more specific account of the line advance mechanism is given in Chapter Four; for the present purpose, the foregoing describes the end result.) The two remaining code statements change the point size to **18** and the body lead to **36.** The eighteen point leading provides the specified 1½ picas between lines of the chapter title when the lines are broken by new line codes.

Following the new line code which initiates the final title line, the code **bl,42** changes body lead to 42 points. This value, in combination with point size **18,** will yield 24 points, or two picas, between the final title line and the opening line of chapter text. Following the final line of the title, the codes **[nl;ps,10;bl,12;tf,18]** change the size and font to 10/12 Video Gael, Size Range II, for the chapter opening.

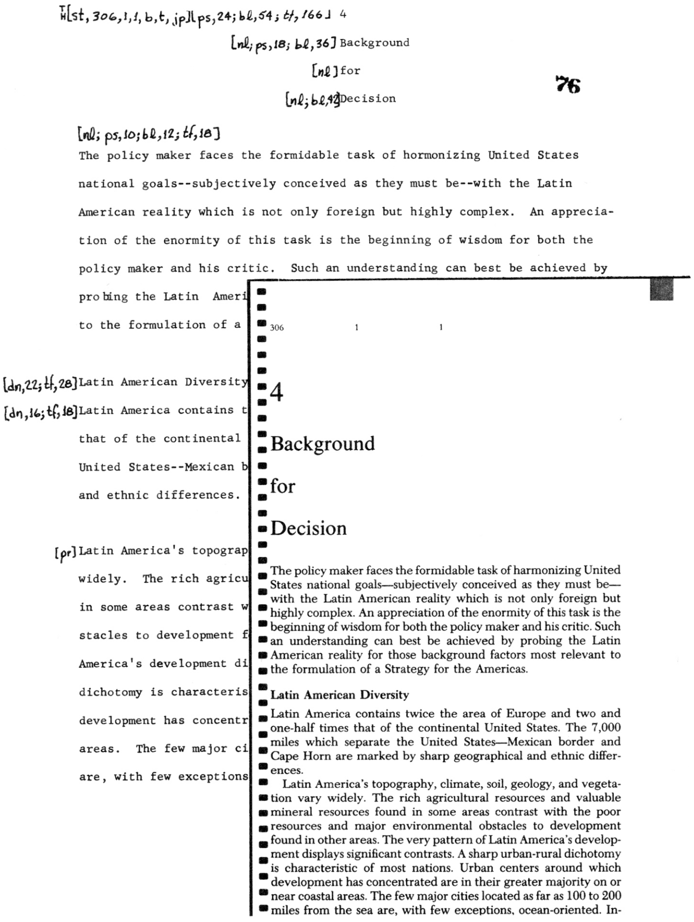
From this point, the processor will set chapter text as long as it appears in input or until some additional control signaling typographic variation appears. Typical typographic variations are side heads of one or several levels, secondary or extract text material, footnotes, or the next chapter opening. Side heads and secondary text usually require changes in size of type, and perhaps typeface, and additional spacing. These changes are made in the input stream in the same way used in setting the chapter heading.

In the sample shown in Figure 3.3, the first level side head is preceded by twelve points of white space and is set in 10 point Video Gael Bold. The baseline of the line following a *down* command is coincident with the vertical position specified by the down. Down twenty-two provides the amount of vertical space specified: the desired twelve points plus ten for the point size of the first level side head. Down sixteen provides six points of white space below the first level side head and allows ten points for the text that follows.

In the "galley-like" sample, footnotes may be treated in a way similar to that used in the preparation of the traditional galley. If the book were being typeset in metal using a linecasting machine, the primary text and footnote text would be set on two different machines, or at least using two different magazines on the typesetting machine, because of the difference in point size. The "maker-up", the person in the composing room who puts the galley together, would simply insert the footnote lines between the text line which bears the footnote citation and the line that follows. In the sample, the codes to terminate the text line and to change point size to 8/10 are placed on the manuscript just following the citation in text.

In the hot metal case, the line which has the footnote citation would be a complete, justified line. In the PAGE–1 case, the footnote text must be inserted immediately following the citation in text. The footnote citation in primary text becomes the last primary text character set until the footnote text has been disposed

Figure 3.3
Galley



TH[st,306,1,1,b,t,jp][ps,24;bl,54;tf,166] 4
[nl;ps,18;bl,36]Background
[nl]for
[nl;bl,42]Decision
76

[nl;ps,10;bl,12;tf,18]
The policy maker faces the formidable task of hormonizing United States
national goals--subjectively conceived as they must be--with the Latin
American reality which is not only foreign but highly complex. An apprecia-
tion of the enormity of this task is the beginning of wisdom for both the
policy maker and his critic. Such an understanding can best be achieved by
pro bing the Latin Ameri
to the formulation of a

[dn,22;tf,28]Latin American Diversity
[dn,16;tf,18]Latin America contains t
that of the continental
United States--Mexican b
and ethnic differences.

[pr]Latin America's topograp
widely. The rich agricu
in some areas contrast w
stacles to development f
America's development di
dichotomy is characteris
development has concentr
areas. The few major ci
are, with few exceptions

306 1 1

4

Background

for

Decision

The policy maker faces the formidable task of harmonizing United States national goals—subjectively conceived as they must be—with the Latin American reality which is not only foreign but highly complex. An appreciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

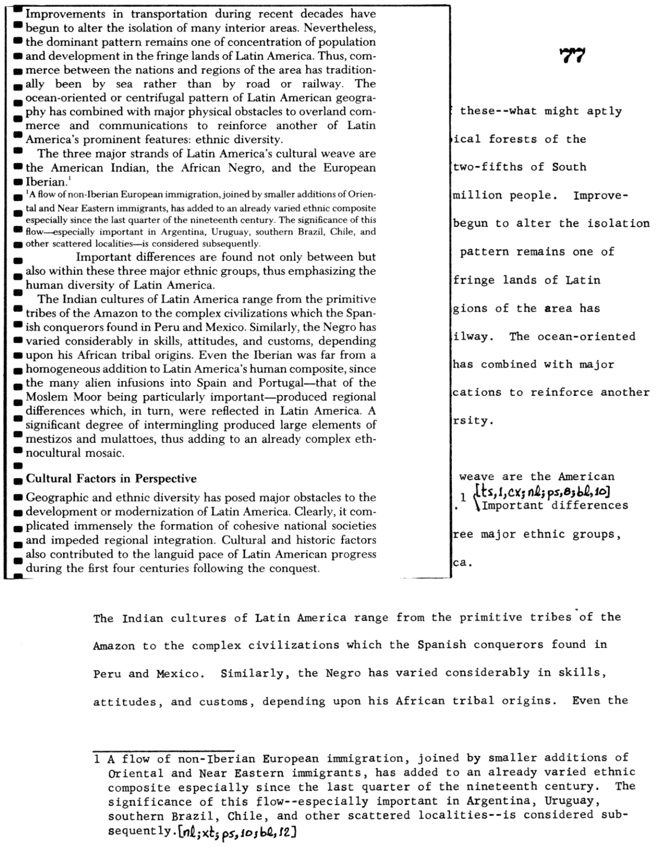
Latin American Diversity

Latin America contains twice the area of Europe and two and one-half times that of the continental United States. The 7,000 miles which separate the United States—Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

Latin America's topography, climate, soil, geology, and vegetation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's development displays significant contrasts. A sharp urban-rural dichotomy is characteristic of most nations. Urban centers around which development has concentrated are in their greater majority on or near coastal areas. The few major cities located as far as 100 to 200 miles from the sea are, with few exceptions, ocean-oriented. In-of. In order to preserve the line breaks, and thereby the total number of lines in which the paragraph which has a footnote citation will set, a tab stop can be established following the citation and before the changes required to set the footnote. The resumption of primary text, following the footnote can then start at the tab stop. The result is two partial lines; one immediately precedes the footnote and ends with the citation and the other follows the footnote and starts at the tab stop. Together they form a single text line. Note that in some circumstances the result may be erroneous since the words of the first fragment are separated by the *basic interword space;* the fragment has not been justified. The result when the entire line is justified may not be equivalent to the sum of the two fragments and in some cases the difference may affect the line break and possibly the number of lines in which the paragraph sets.

The code string to interrupt text and prepare for the footnote is **[ts,1,cx;nl;ps,8;bl,10]**; it specifies that a tab stop numbered one is to be established at the horizontal position at the time the code is encountered. Following the footnote, the codes to resume primary text are **[nl;xt,1;ps,10;bl,12]**; they cause a new line and specify that it is to start at the tab position defined before the footnote. These code strings are shown on the manuscript of Figure 3.4. The discussion of foot-

Figure 3.4
Galley footnote



Improvements in transportation during recent decades have begun to alter the isolation of many interior areas. Nevertheless, the dominant pattern remains one of concentration of population and development in the fringe lands of Latin America. Thus, commerce between the nations and regions of the area has traditionally been by sea rather than by road or railway. The ocean-oriented or centrifugal pattern of Latin American geography has combined with major physical obstacles to overland commerce and communications to reinforce another of Latin America's prominent features: ethnic diversity.

The three major strands of Latin America's cultural weave are the American Indian, the African Negro, and the European Iberian.[1]

[1]A flow of non-Iberian European immigration, joined by smaller additions of Oriental and Near Eastern immigrants, has added to an already varied ethnic composite especially since the last quarter of the nineteenth century. The significance of this flow—especially important in Argentina, Uruguay, southern Brazil, Chile, and other scattered localities—is considered subsequently.

Important differences are found not only between but also within these three major ethnic groups, thus emphasizing the human diversity of Latin America.

The Indian cultures of Latin America range from the primitive tribes of the Amazon to the complex civilizations which the Spanish conquerors found in Peru and Mexico. Similarly, the Negro has varied considerably in skills, attitudes, and customs, depending upon his African tribal origins. Even the Iberian was far from a homogeneous addition to Latin America's human composite, since the many alien infusions into Spain and Portugal—that of the Moslem Moor being particularly important—produced regional differences which, in turn, were reflected in Latin America. A significant degree of intermingling produced large elements of mestizos and mulattoes, thus adding to an already complex ethnocultural mosaic.

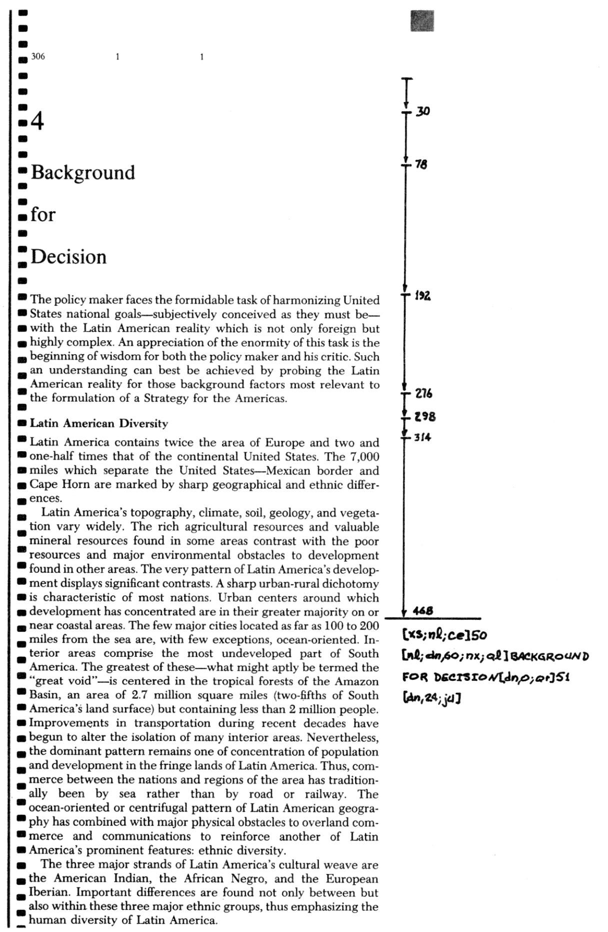
Cultural Factors in Perspective

Geographic and ethnic diversity has posed major obstacles to the development or modernization of Latin America. Clearly, it complicated immensely the formation of cohesive national societies and impeded regional integration. Cultural and historic factors also contributed to the languid pace of Latin American progress during the first four centuries following the conquest.

```
 these--what might aptly
ical forests of the
two-fifths of South
million people.  Improve-
begun to alter the isolation
 pattern remains one of
fringe lands of Latin
gions of the area has
ilway.  The ocean-oriented
has combined with major
cations to reinforce another
rsity.

 weave are the American
 1 [ts,1,cx;nl;ps,8;bl,10]
.  Important differences
ree major ethnic groups,
ca.
```

```
The Indian cultures of Latin America range from the primitive tribes of the

Amazon to the complex civilizations which the Spanish conquerors found in

Peru and Mexico.  Similarly, the Negro has varied considerably in skills,

attitudes, and customs, depending upon his African tribal origins.  Even the

_______________
1 A flow of non-Iberian European immigration, joined by smaller additions of
  Oriental and Near Eastern immigrants, has added to an already varied ethnic
  composite especially since the last quarter of the nineteenth century.  The
  significance of this flow--especially important in Argentina, Uruguay,
  southern Brazil, Chile, and other scattered localities--is considered sub-
  sequently.[nl;xt;ps,10;bl,12]
```

Figure 3.5
Page break


30
78
192
276
298
314
468

306 1 1

4

Background

for

Decision

The policy maker faces the formidable task of harmonizing United States national goals—subjectively conceived as they must be—with the Latin American reality which is not only foreign but highly complex. An appreciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

Latin American Diversity

Latin America contains twice the area of Europe and two and one-half times that of the continental United States. The 7,000 miles which separate the United States—Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

Latin America's topography, climate, soil, geology, and vegetation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's development displays significant contrasts. A sharp urban-rural dichotomy is characteristic of most nations. Urban centers around which development has concentrated are in their greater majority on or near coastal areas. The few major cities located as far as 100 to 200 miles from the sea are, with few exceptions, ocean-oriented. Interior areas comprise the most undeveloped part of South America. The greatest of these—what might aptly be termed the "great void"—is centered in the tropical forests of the Amazon Basin, an area of 2.7 million square miles (two-fifths of South America's land surface) but containing less than 2 million people. Improvements in transportation during recent decades have begun to alter the isolation of many interior areas. Nevertheless, the dominant pattern remains one of concentration of population and development in the fringe lands of Latin America. Thus, commerce between the nations and regions of the area has traditionally been by sea rather than by road or railway. The ocean-oriented or centrifugal pattern of Latin American geography has combined with major physical obstacles to overland commerce and communications to reinforce another of Latin America's prominent features: ethnic diversity.

The three major strands of Latin America's cultural weave are the American Indian, the African Negro, and the European Iberian. Important differences are found not only between but also within these three major ethnic groups, thus emphasizing the human diversity of Latin America.

[xs;nl;ce]50
[nl;dn,60;nx;ql]BACKGROUND
FOR DECISION[dn,0;qr]51
[dn,24;jd]

notes will now be interrupted; it will be picked up again in Chapter Four.

Page proofs

The foregoing paragraphs have presented the preparation of manuscript material for typesetting galleys for a book. The analogy can be carried a step further to generate page proofs. The discussion which follows is by no means pagination; the intent is to describe the operation of a few codes and to introduce the functions which are required for pagination. In the hot metal case, page breaks are marked on the galley proof. Subsequently the maker-up dismantles the galley and reassembles it with folios and running heads. At this stage of the process, the maker-up may increase or decrease spacing around side heads to assure specified page depth. He also puts footnotes at the bottom of the pages on which they occur and breaks them between pages if this is necessary.

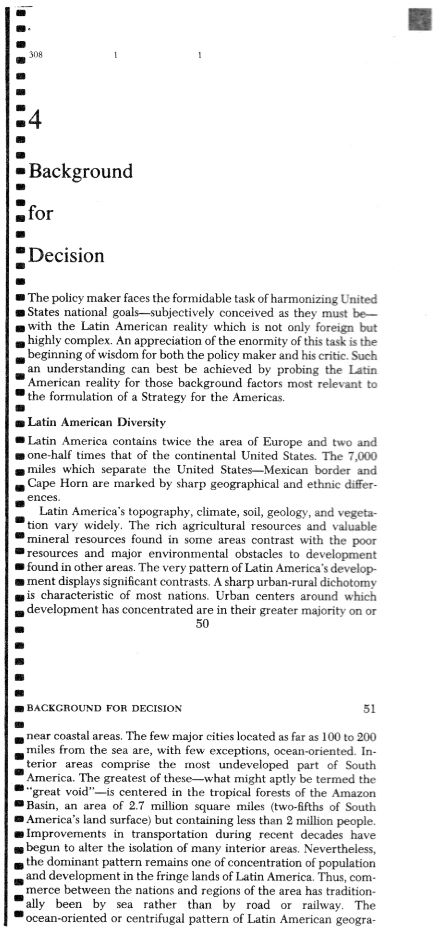
Assume that the book specification calls for 2½ picas above the chapter number, that the page number is set in ten point Video Gael, centered at the bottom of the page with two points above, and that pages are 40 picas deep. Figure 3.5 is the galley of Figure 3.3 marked to show depth at several points. First consider the closing of the chapter opening page and then the initiation of the succeeding page. It happens in this sample that the depth to the bottom of the point size frame of the line above the page break is 468 points. The ten point *drop folio* with two points above requires twelve points and total depth of the opening page is 480 points or the required 40 picas.

Since the drop folio is to be set in the same point size and typeface as chapter text and since the two points required above it is equal to primary text leading, it is necessary only to terminate the text line and change the justification mode to *center* in order to typeset the drop folio. The codes to do this are **[nl;ce].** Following these codes, the page number is added and the line is closed with a new line code. With a single omission, the chapter opening page now coded will meet the style specification.

The final line before the page break is a justified line. That is to say, the word spaces have been increased or decreased from the basic interword space by the hyphenation and justification program. If they were decreased, all will be well; but if they were increased, the line will be short when the same line is terminated by the new line code. It is possible to avoid this possibility by using the *expand-to-fill-measure* code, **xs**. Following use of **xs**, all lines terminated by the new line code (or by any forced line break) will be expanded to fill measure; that is, justified. The drop folio line is concluded with a new line code and a space between the two pages is provided by a down sixty.

Now consider the continuation page. Assume that the book is to have a running head at the top consisting of either the book title or the chapter title (odd or even numbered page respectively) set to the inside margin and the page number set to the outside margin. The running head is to be set in ten point Video Gael and there is fourteen points of white space separating the running head from text. If the number of the chapter opening page is 50, it is a left-hand page. The

Figure 3.6
Page proof



308 1 1

4

Background

for

Decision

The policy maker faces the formidable task of harmonizing United States national goals—subjectively conceived as they must be—with the Latin American reality which is not only foreign but highly complex. An appreciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

Latin American Diversity

Latin America contains twice the area of Europe and two and one-half times that of the continental United States. The 7,000 miles which separate the United States—Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

Latin America's topography, climate, soil, geology, and vegetation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's development displays significant contrasts. A sharp urban-rural dichotomy is characteristic of most nations. Urban centers around which development has concentrated are in their greater majority on or

50

BACKGROUND FOR DECISION 51

near coastal areas. The few major cities located as far as 100 to 200 miles from the sea are, with few exceptions, ocean-oriented. Interior areas comprise the most undeveloped part of South America. The greatest of these—what might aptly be termed the "great void"—is centered in the tropical forests of the Amazon Basin, an area of 2.7 million square miles (two-fifths of South America's land surface) but containing less than 2 million people. Improvements in transportation during recent decades have begun to alter the isolation of many interior areas. Nevertheless, the dominant pattern remains one of concentration of population and development in the fringe lands of Latin America. Thus, commerce between the nations and regions of the area has traditionally been by sea rather than by road or railway. The ocean-oriented or centrifugal pattern of Latin American geogracontinuation page will be a right-hand page; the chapter title will be set to the left-hand margin (the inside) and the page number will be set to the right-hand (the outside) margin. See Figure 3.6.

The running head may be considered to be two "lines" which have a common baseline. The chapter title may be set *flush left* and the line terminated by a *down zero* command to effect the flush left action while providing for the rest of the running head to be set on the same baseline. The page number then may be set *flush right* and the line terminated with a *down twenty-four* to close out the running head and to provide fourteen points between running head and the continuing text.

Following the down sixty and prior to the running head, the expand-to-fill-measure mode must be canceled and, following the running head, the *justify horizontally* mode must be reestablished before the continuing text is resumed. The code to cancel the expand-to-fill-measure mode is **[nx]** and the justify code is **[ju]**. The required code strings become:

- Prepare for drop folio,
  [xs;nl;ce]50
- Page break and prepare for chapter title,
  [nl;dn,60;nx;ql]BACKGROUND FOR DECISION
- Terminate page number line and
  prepare for page number,
  [dn,0;qr]51
- Terminate running head and prepare for text,
  [dn,24;ju]

These codes, together with text; that is, the page numbers and the running head text are shown on the galley proof in Figure 3.5 and the page proofs are shown in Figure 3.6.

The development to this point might rightly be termed "electronic hand composition"; it is tedious in the extreme and certainly not a viable method of typesetting. Further, the second page to be made up contains a footnote which cannot be handled with the codes used so far. At this point further work on the sample will be interrupted in favor of a further discussion of the PAGE–1 répertoire of codes.

4

FURTHER PAGE–1 CODES

THE REMAINDER of the codes in the PAGE–1 instruction complement will now be introduced. Their functions will be described in prelude to the serious development of *typographic format statements* in Part III.

Print region

PAGE–1 may be viewed as setting text within the edges of rectangular sheets of film that are called *print regions.* A printed page is, in general, analogous to a print region; each of the samples produced in Chapter 3 is a single print region.

The 70|800 family of VideoComps has a usable width between sprocket holes of 70 picas; in cut film mode they will produce sheets or pages up to 22 inches deep. PAGE–1, in fact, sets text in *blocks* whose positions on the print region are specified as desired and changed during composition as the text on any given page may dictate. Block location may be changed at any time during the composition of a print region; its location at the time the print region is closed is the effective location for that page. (The method used in Chapter 7 for setting footnotes is one example of the need to change block location as composition procedes.)

The print region has a reference point or origin for the purpose of block location. This reference point is on the inside edge of the sprocket holes on the left side of the film as it is viewed and at a point below the top edge of the piece of cut film. The reference point is below the top edge of the film to allow for handling and for placement of several identification numbers. The start of a new print region (and the close of the one just completed) is specified by the code **[pg]**.

Text blocks

The 70|800 Family of VideoComps set a line of type on stationary film and then advance the film for the next line. The concept of *text blocks* provides the capability to handle the out-of-sequence typesetting of some types of input material. For example, to set a two-column page, the VideoComp must be given "lines" comprising the corresponding column-one and column-two lines. The sense of the text on the page—and the order of the text in input—starts at the top of the first column and ends at the bottom of the second. A footnote appears at the bottom of the printed page, but in input it appears immediately following the citation in primary text. The text block concept provides for this out-of-order typesetting.

A text block is an area in which characters are set from left to right within a line and line by line downward. The location, or position, of a text block on a print region is specified by two codes which relate its origin (its upper left corner) to the origin of the print region:

X-coordinate:—	**[bx,1,X1]**
Y-coordinate:—	**[by,1,Y1]**

The first statement specifies the origin of block one to be **X1** points rightward from, and the second specifies it to be **Y1** points below, the origin of the print region; both distances are specified in points.

The *start block* code, **[sb,1]**, specifies that the manuscript material which follows is to be set in block number one. A text block has a number of *attributes* which specify such things as the font and size in which text is to be set, the line length and column depth, the hyphenation and justification conventions, and boundary procedures which may be used to effect page make-up.

Text block dimensions

Block size is specified by four boundaries. The top and bottom boundaries are specified in points below block origin and the left and right boundaries are specified in points rightward from block origin. The codes are:

Top:— **[tb,0]** Bottom:— **[bb,y2]**
Left:— **[lb,0]** Right:— **[rb,x2]**

In general, left and top boundaries will be zero; that is, left and top boundaries will pass through the block origin. The left boundary may be moved rightward temporarily, for example, to provide a standing indention for secondary text. Left and right boundary changes may be stated at any time but they will not be effective unless or until **cx** is zero. (The code, **cx**, is a system variable which measures the *current horizontal setting position* with respect to the left boundary.) Left and right boundary changes are effective only at the start of a new line before any text has been set. Correspondingly, a top boundary change may be stated at any time but it is effective only if or when **cy** is zero. (The code, **cy**, is a system variable which measures the *current vertical setting position* with respect to the block top boundary.) That is, a top boundary change can only be effective at the start of a block before any text has been set.

The right boundary, together with the left boundary, specifies column measure; that is, the length of the line to be set. The bottom boundary specifies the depth of the text column; it need be specified only if bottom boundary procedures are used. Changes in bottom boundary may be stated at any time; the changes are effective immediately.

Line composition

Manuscript material must be divided into lines of measure specified by the difference between the left and right boundaries of the block in which setting is in progress. The input text may be set justified (lines of equal length) or it may be set unjustified.

The composition of justified lines is specified by the code **[ju]** which states the text is to be set justified. Justification is further defined to allow or inhibit the use of hyphenation in the process. If hyphenation is allowed, it may be with or without use of the exception dictionary. PAGE–1 provides hyphenation by an algorithmic process with an exception dictionary which contains those words that are improperly hyphenated by the hyphenation algorithm. (The exception dictionary may also contain words for which the user prefers hyphenation points other than those provided by the hyphenation algorithm.) The codes are:

Horizontal justification:— **[ju]**
Inhibit hyphenation:— **[nh]**
Hyphenation allowed:— **[hy]**
Use dictionary:— **[xd]**
No dictionary:— **[nd]**

Justification is effected by varying the size of the spaces between words and additionally, perhaps, by hyphenation. During line composition, wordspaces are tentatively assigned the value of the *basic interword space*. When a word would overset the right boundary, the accumulated widths of the interword spaces on the line is decreased—subject to the lower limit assigned to the *minimum interword space*—in an attempt to fit the entire word on the current line. Failing this, the accumulated widths of the interword spaces is increased—subject to the upper limit assigned to the *maximum interword space*—in an attempt to push the entire word to the next line. If these attempts fail and if hyphenation is inhibited, the width of the interword spaces are at this point increased without limit to push the entire word to the next line. If hyphenation is permitted and the attempt to push the entire word to the next line has failed, the word will be hyphenated by the logical hyphenation program and, if allowed, the exception dictionary.

Input text may also be set unjustified. In the non-justify modes, words are separated by the basic interword space. Each time a word which would overset the right boundary is encountered, the line is terminated at the preceding interword space and a new line is started with the word which caused overset. There are three non-justify modes: *flush left*—in which all unused space is inserted at the end of the line giving a "ragged right" output column; *center horizontally*—in which unused space is inserted half at the beginning and half at the end of the line giving a "ragged left and right" output column; and *flush right*—in which all unused space is inserted at the beginning of the line giving a "ragged left" output column. The non-justify codes are:

Flush left:— **[ql]**
Center horizontally:— **[ce]**
Flush right:— **[qr]**

Occasional typographic requirements specify that all lines fill measure even though they be ended by forced line breaks. The *expand-to-fill-measure* code serves this need. Expand-to-fill-measure is mutually exclusive with the three non-justify modes. In justify mode, then, the code **[xs]** causes the interword spaces in all lines which are terminated by a forced line break to be expanded such that the line fills measure. The code may be stated with a parameter to specify the action only if the remaining space (to be absorbed by the expansion) is less than the number of points specified by the parameter. For example, if the statement **[xs,24]** has been issued, only those lines which are terminated within 24 points of the right boundary will be expanded. The rest will be set flush left. This mode is cancelled by the code **[nx]**.

PAGE–1 provides fifteen tab stops. The position of a tab stop is specified in points rightward from the left boundary with a statement of the form **[ts,8,150]** which positions tab stop number eight 150 points to the right of the left boundary. If the left boundary is moved, the tab stop moves with it. The statement **[rt]** sets the position of all fifteen tab stops to zero thereby cancelling those which have been defined.

Text may be set from tabs, flush right to tabs, or centered between tabs. The execute tab statements take the form:

Tab left:—	[xt]	or	[xt,3]
Tab center:—	[xc]	or	[xc,4]
Tab right:—	[xr]	or	[xr,5]

If no tab number is explicitly stated, the horizontal setting position will be advanced to the first tab stop position rightward from the then current setting position. If a tab number is stated, the horizontal setting position will be moved to the position specified by the tab number whether it is to the right or the left of the then current setting position.

The *tab left* statement causes the text material which follows—before another execute tab statement or a forced line break—to be set flush left to the tab stop cited or implied.

The *tab center* statement causes the text material which follows—before another execute tab statement or a forced line break—to be set centered between the tab stop cited (or implied) and:

- The tab cited by a terminating tab statement with a parameter,
- The next larger defined tab position following **cx** if the terminating tab statement has no parameter or if the tab position specified by the terminating command is to the left of **cx**, or
- The right boundary if the terminating statement is a forced line break.

The *tab right* statement causes the text material which follows—before another execute tab statement or a forced line break—to be set flush right to:

- The tab cited by a terminating tab statement with a parameter,
- The next larger defined tab position following **cx** if the terminating tab statement has no parameter or if the tab position specified by the terminating command is to the left of **cx**, or
- The right boundary if the terminating statement is a forced line break.

Vertical space

Lines of text set in a block start at the top boundary; one might think of them as being "flushed up". If an entire text column is completed and no change from straight primary text has been encountered, the final line should just reach the bottom boundary of the text block. If, however, there has been a side head with added space above and below or a few lines of secondary text which was set in a smaller point size, it is likely that the last line in the text column falls short of reaching the bottom boundary. In manual page make-up such situations are generally handled by adding a little lead above and/or below the side head or the secondary text such that the finished page is of the specified depth.

PAGE–1 provides two ways of handling this need. The first is called *justify vertically* and results in the expansion of interline lead such that the ascenders of the first line are flush against the top boundary and the descenders of the last line are flush to the bottom boundary. The code is **[jv]**. Surplus space, that between the descenders of the last line in the block and the bottom boundary, is allocated to the interline spaces in equal increments starting with the space between the first and second lines in the block. (Space above lines initiated by the *down* commands is not subject to vertical justification—see discussion under forced line breaks.)

The alternative method is to allocate the surplus space around side heads and secondary text. This method requires that space around side heads and secondary text be specified by the *justifiable down* statement. (See discussion under forced line breaks.) Using this method, surplus space is allocated equally among the lines initiated by the justifiable down statement. The code is **[jd]**. The codes **[jv]** and **[jd]** may be used in the same block at the same time. If there are no justifiable down statements in the block, the **jd** has no effect. The text material set in a block can be *cen-*

tered vertically between the top and bottom boundaries of the block or *flushed down* to the bottom boundary. The codes are **[cv]** and **[qd]** respectively. If none of the vertical justification codes are invoked, the top line is set flush to the top boundary and the interline lead is not altered.

These four vertical space codes are not "standing procedures" as are the analogous horizontal space codes. They are executed once when the block is terminated and then discarded.

Boundary procedures

The boundary procedures enable the user "program" to regain control when, in the course of building lines or blocks of text, the processor encounters a right or bottom boundary. These procedures enable the user program to call and execute procedures at points in the text stream which cannot be anticipated in the input data. For example, in Figure 3.3, it was necessary to see the galley before the page break could be determined.

PAGE–1 provides two classes of boundary procedures: *at boundary* and *when boundary.* An at boundary procedure is executed once at the first encounter of the corresponding boundary after the procedure is placed in force and is then discarded. A when boundary procedure is executed whenever its corresponding boundary is encountered, and remains in force for succeeding encounters of the boundary, with the single exception that if at boundary and when boundary procedures for the same boundary are present simultaneously, the at boundary procedure takes precedence. It will be executed and the when boundary procedure will not.

The right boundary is said to have been encountered when an additional word will not fit on the line being composed. The overset condition is detected and the appropriate *at right* or *when right* procedure, if either or both are present, is retrieved and executed. Right boundary procedures are, in effect, executed *at the left boundary;* that is, after the line advance and the update of **cx** and **cy** have been accomplished.

Prior to starting each new line of text, the processor checks to be sure that another line will fit. If the space remaining is less than the point size in force, the bottom boundary is said to have been encountered. The processor then checks for the presence of a bottom boundary procedure and, if there is one, executes it. The boundary procedures take the following form:

```
[xx[[code to be executed ]]]
[xx[[code]text to be set ]]
```

in which **xx** represents any of the four boundary operators. Note that the nest level determines the nature of the material in the procedure. Odd numbered nest levels are control to be executed and even numbered nest levels are text which is to be set. If text is to be set "in a bottom boundary procedure", it must be set in a different block or, if in the same block, the bottom boundary must first be extended to provide space for additional text.

Consider again the page proof example from Chapter Three; bottom boundary procedures may be used to determine the page break and to specify the appropriate action. Modify the code string from Figure 3.3 to put the manuscript text explicitly in block one, add a bottom boundary, and an at bottom procedure:

```
[sb,1;bb,468;ps,24;bl,48;tf,166;dn,30;
 ab[[bb,478;nl;ce]50[nl;dn,60]BACKGROUND FOR DECISION
    [dn,0;qr]51[dn,24;ju]]]]
```

This statement can be placed at the beginning of the text without regard for the point in text at which the page breaks. Not pagination yet; but a step along the way.

Interrupt and continue block

Since the PAGE–1 system accepts input data in one continuous stream of mixed manuscript data and control, it follows that the only place to enter the words which comprise a footnote is immediately following its citation in text. (Unless, of course, all footnotes are collected and set at the close of the chapter.) Assuming that the footnotes are to be set at the bottom of their citation page and that they are entered as suggested above, they must be accommodated when they appear in the input stream.

Given the text:footnote:text data stream, it is necessary to interrupt the processing of primary text and handle the footnote when it appears. The *interrupt block* facility does this. PAGE–1 has the capacity to handle two blocks concurrently. It is convenient to think of a *primary* block in which most of the input manuscript material is being set and an *interrupting* block in which the "out of sequence" material is to be set. The interrupt block code, **[ib]**, causes the processor to store the attributes of the primary block and many details regarding the point at which the block is interrupted in order that composition may be resumed at the point of interruption when the continue block code, **[cb]**, is encountered.

After storing the attributes and other data for the primary block, the processor calls the stored attributes and data regarding the interrupting block in preparation for composing the out of sequence material. The first time that **[ib]** appears in a job, there is no interrupting block yet identified; the processor will in this one case assign the attributes of the primary block to the interrupting block. The user must follow the interrupt block code with a start block in order to identify the interrupting block and he may also reassign any block attributes which are different from their counterparts in the primary block; font, size, and justification mode for example.

When the continue block code is encountered, the process just outlined is reversed. The attributes and details of composition in the interrupting block are stored and those of the primary block are recalled. Succeeding interrupt block codes need not be followed by the start block code unless the identity of the interrupting block is to be changed.

The footnote of Figure 3.4 can now be handled by the data stream which follows:

[ib;sb,2;ps,8;bl,10] *footnote* **[nl;cb;bx,2,300;by,2,690]**.

The codes to establish the origin of block two are added only in the interest of having the footnote appear adjacent to its citation in the illustration. The positioning of the footnote is considered in detail in the case studies of Chapter Seven.

Forced line breaks

Forced line breaks, those which the user requests at specific points in the manuscript (and in contrast to natural line breaks which result from right boundary encounter), appear in many situations in typesetting. Headings, indexes, tables of contents, in addition to the many paragraph breaks which occur in straight text, are but a few examples. PAGE–1 has two classes of forced line breaks, those which simply start a new line and those which specify a sink before the new line is started.

New line codes

The *new line* code instructs the processor to terminate the current line and to start a new line beginning at the left boundary and spaced downward by a function of the point size and body lead in force. A new line code may be issued with a parameter to specify the number of blank lines desired between the line being terminated and the one being initiated. If the new line is encountered immediately following a natural line break (that is, when **cx** is zero), the processor assumes that another new line—resulting in one blank line—is not desired and ignores the explicitly stated new line.

The *new paragraph* code is similar in every respect but one to the new line code. The line initiated is spaced from the left boundary by the amount of the standard paragraph indention which is defined via the typographic variable **pi**.

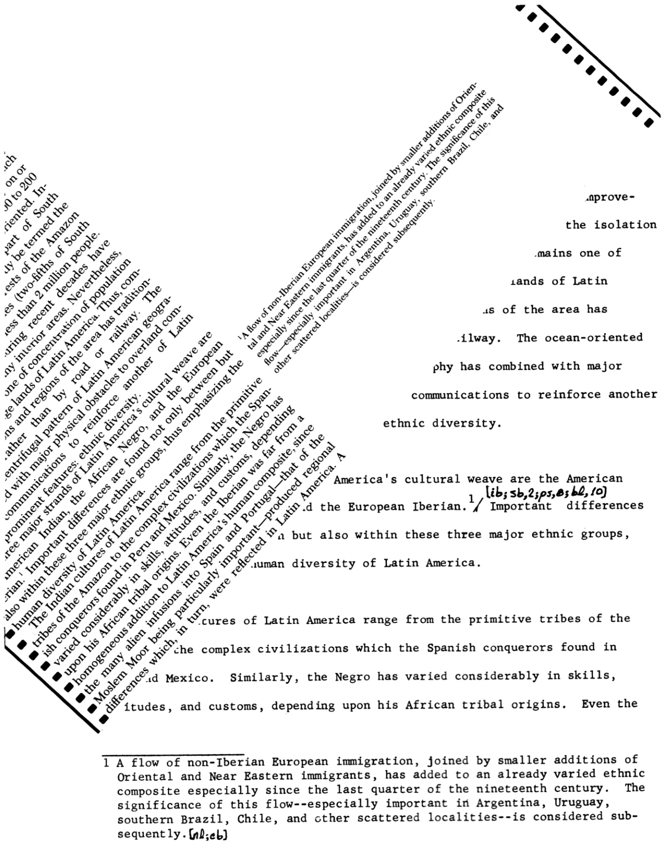
The new line and new paragraph codes take the following form:

New line:—	**[nl]** or	**[nl**,parameter]
New paragraph:—	**[pr]** or	**[pr**,parameter]

Down codes

The *fixed down* code instructs the processor to terminate the current line and to initiate a new line starting at the left boundary and spaced below the current

Figure 4.1
The interrupting block



.mprove-
the isolation
.mains one of
.ands of Latin
.s of the area has
.ilway. The ocean-oriented
phy has combined with major
communications to reinforce another
ethnic diversity.

America's cultural weave are the American
[ib;sb,2;ps,8;bl,10]
.d the European Iberian.1/ Important differences
.n but also within these three major ethnic groups,
.uman diversity of Latin America.

.tures of Latin America range from the primitive tribes of the
.he complex civilizations which the Spanish conquerors found in
.d Mexico. Similarly, the Negro has varied considerably in skills,
.itudes, and customs, depending upon his African tribal origins. Even the

1 A flow of non-Iberian European immigration, joined by smaller additions of Oriental and Near Eastern immigrants, has added to an already varied ethnic composite especially since the last quarter of the nineteenth century. The significance of this flow--especially important in Argentina, Uruguay, southern Brazil, Chile, and other scattered localities--is considered subsequently. [nl;cb]

line by the number of points specified in the parameter. A line initiated by the fixed down code is not subject to vertical justification by either the **jv** or **jd** command.

The *justifiable down* code is similar to the fixed down. The difference is that the parameter is interpreted as a tentative sink subject to increase if the **jd** type of vertical justification is imposed when the block is closed. A line initiated by the justifiable down code is not subject to vertical justification by the **jv** command.

Both down codes must be issued with a parameter to express the amount of the sink:

Fixed down:— [**dn**,parameter]
Justifiable down:— [**dj**,parameter]

Variables

The term variable refers to a subset of the PAGE–1 primitive codes which is unique in that its members may be used as parameters or qualifiers to the other primitive codes and in arithmetic and conditional procedures. It is convenient to think of variables in three classes:

- *Typographic variables* which are block oriented—for example, point size may have a different value in the primary and interrupting blocks;
- *Global variables* which are job oriented—page number, for example, has only one value at any given time; and
- *System data*—codes to which the user may not assign values, but which provide him the facility to sample horizontal and vertical setting position.

Typographic variables

Most of the typographic variables have already been introduced. In general they specify design considerations such as typeface and size, text block dimensions and standard spacing considerations. The typographic variables are:

Top boundary:— **tb** *Bottom boundary:—* **bb**
Left boundary:— **lb** *Right boundary:—* **rb**
Point size:— **ps** *Body lead:—* **bl**
Set width:— **sw** *Typeface:—* **tf**
Page width:— **pw** *Page depth:—* **pd**
Minimum interword space:— **mi**
Basic interword space:— **bs**
Maximum interword space:— **mx**

Only two new variables appear in this list. Page width and depth are convenient variables to use in writing code strings; they specify nothing in the sense that the other typographic variables do. Rather, they are convenient short-hand variables for use in writing control statements.

Global variables

The global variables are variables in the accepted sense except that their number is limited and they have explicit "names". Three of the global variables, *page number, footnote number,* and *standard paragraph indention,* have specific two-character names as do the typographic variables. In addition to these, the system has 201 *general variables* and nine *indirect variables.* The general variables are simply numbered zero through 200 and the indirect variables one through nine. General variable zero is associated with the *execute* code and has a well defined use in that context. The remainder of the general variables and all of the indirect variables provide the capability to identify and establish specific values for meaningful variable quantities. Then, as with all of the variables, they are used as parameters to other PAGE–1 codes and arithmetic and conditional statements. The global variables are:

Page number:— **pn**
Footnote counter:— **fn**
Standard paragraph indention:— **pi**
General variable:— **gv***n* $0 \leqslant n \leqslant 200$
Indirect variable:— **iv***n* $1 \leqslant n \leqslant 9$

System data

The three codes in this class: *current horizontal setting position, current vertical setting position,* and *current character,* may be used as parameters to other PAGE–1 codes, but the user may not assign values to them. They provide the facility to "interrogate" position in a line or block and to involve that information in ways that will effect subsequent composition activity.

Current horizontal setting position is measured from the left boundary to the right edge of the last character set before the code is encountered. The processor maintains this value in internal units but delivers it, when **cx** appears as a parameter, in points.

The current vertical setting position is measured in points below the Y-coordinate of block origin. Its value, when **cy** appears as a parameter, depends upon further circumstances. If the line being set when **cy** is encountered was started by a natural line break, by a new line code, or by a new paragraph code, the measurement is to the bottom edge—including leading—of the last complete line set. If, however, the partial line was started by one of the down commands, the measurement is to the bottom edge—excluding leading—of the line being set. If interrogated after bottom boundary encounter; that is, in a bottom boundary procedure, **cy** is measured to the bottom edge—excluding leading—of the last line in the block.

Current character, when interrogated, delivers the decimal representation of the internal code for the most recently set character. Its primary use is in conditional procedures within boundary procedures, which are to be executed if the last character set was a specific character; for example, a hyphen. The code for current character is **cc**.

Arithmetic operations

Six arithmetic operators and two "compare and store" operators accommodate the requirements of typographic compositiona and page make-up. The arithmetic statement takes the form:

[xx,variable,parameter,parameter]

in which **xx** is any of the eight arithmetic operators. The term *variable* refers to those codes defined earlier as variables and the term parameter includes integer constants in addition to the variables just tabulated.

The arithmetic statements are:

Sum	[**ad**,variable,parameter,parameter]
Difference	[**df**,variable,parameter,parameter]
Product	[**ml**,variable,parameter,parameter]
Quotient	[**qo**,variable,parameter,parameter]
Quotient (raised)	[**rq**,variable,parameter,parameter]
Remainder	[**rm**,variable,parameter,parameter]
Greater of	[**gr**,variable,parameter,parameter]
Lesser of	[**ls**,variable,parameter,parameter]

In which the first parameter is the augend and the second the addend for a sum, the first is the minuend and the second the subtrahend for a difference, the first is the multiplicand and the second the multiplier for a product, and the first is the dividend and the second the divisor for the three quotient statements. The compare statements simply result in a comparison of the numeric values of the two parameters followed by assignment of the greater or lesser of the pair to the variable.

Conditional procedures

Conditional procedures provide the capability to execute strings of control statements, to set text, or to handle any combination of control statements and text on the condition that the specified result of a comparison is satisfied. If the stated condition is not fulfilled, the conditional statement is omitted. The conditional statement takes the form:

[**xx**,parameter,parameter[*text to be set*]]
[**xx**,parameter,parameter[[*code to be executed*]]]
[**xx**,parameter,parameter[[*code*]*text*]]

in which **xx** represents any of the six conditional operators. Note that the nest level determines the nature of the material in the procedure. Odd numbered nest levels are control and even numbered nest levels are text. The conditional procedures are:

[**lt**,a,b[[control]]]	Control is executed if $a < b$
[**le**,a,b[[control]]]	$a \leqslant b$
[**eq**,a,b[[control]]]	$a = b$
[**ne**,a,b[[control]]]	$a \neq b$
[**ge**,a,b[[control]]]	$a \geqslant b$
[**gt**,a,b[[control]]]	$a > b$

A string of conditional procedures should provide for all of the alternatives which can exist in a given test.

In the page proof example of Chapter Three, the left-hand running head was explicitly set with the knowledge that the page was an odd numbered, therefore left hand page. The conditional procedures provide the facility to direct the program to choose between two alternatives. If the page number is divided by two, the remainder which accrues can have either of two values—zero if the page number is even and one if the page number is odd. Two conditional procedures can then provide for selection of the proper running head based on the value of the general variable in which the remainder is stored.

The *set decimal number* code provides the final escape from the need to see the galley. This code takes the form **[dc**,parameter] and will typeset the decimal equivalent of any *variable* (or integer constant) supplied as its parameter. The procedure on page 19 can now be modified using the set decimal number code to typeset the page number, an arithmetic statement to increment the page number counter, and a conditional procedure to select the appropriate running head.

```
...;ab[[bb,478;nl;ce;dc,pn;nl;dn,60;
      ad,pn,pn,1;rm,gv199,pn,2;
      eq,gv199,0[[ql;dc,pn;dn,0;qr]STRATEGY FOR
                THE AMERICAS];
      eq,gv199,1[[ql]BACKGROUND FOR DECISION
                [dn,0;qr;dc,pn]];
      dn,24;ju]
```

This is yet a step closer to pagination because now the statement of running heads and folios appear within the procedure; they are no longer explicit.

Synonyms and format statements

The procedures devised to direct typographic composition are repetitive in nature. In the sample material of Chapter Three, for example, each time a first level side head is encountered, the same code strings written on the manuscript of Figure 3.3 will be used to handle it. While those particular code strings are very short, procedures to direct composition of very complex formats are sometimes quite long.

A great deal of keystroking, and a potential source of catastrophic error, can be avoided by a facility to give names or tags to strings of codes. The code string can then be called for execution by entering its name into input at the appropriate place.

PAGE–1 provides two types of facility to handle material of a repetitive nature—either code strings or text material. *Synonyms* are job oriented and are a part of the data for each job. *Format statements* are global, stored on the system, and are available for call during the processing of any job.

The synonym provides the facility to define two-character, alphanumeric *names* to be synonyms for control strings, text strings, or for any combination of text and control. The first character in the name may be any alphabetic character **t** through **z** and the second may be any digit **1** through **9**, both inclusive. A synonym definition statement takes the form:

```
[sy,xx[text to be set ]]
[sy,xx[[control to be executed ]]]
[sy,xx[[control ]text ]]
```

in which **xx** represents any of the 163 legal synonym names. Note, once more, that nest levels determine the nature of the material which is being named.

Format statements provide the capability to build a library of code strings which are used at a given installation on more than one job. Their storage on the system is a maintenance function. Like synonyms, format statements are given two-character, alphanumeric *labels,* often called *formats.* The first character may be any alphabetic character **a** through **s** and the second any digit **1** through **9**, both inclusive.

Aids to typography

The codes covered to this point are those required in routine composition. The codes following are directed to specific composition problems.

Trial setting

Many occasions in typesetting require a trial set to determine the width, depth, or position on a page in order to set the final version properly. One such requirement arises in the setting of poetry. The longest line of a poem is sometimes centered on the print area and the indention of the other lines is governed by that line.

The trial set facility in PAGE–1 is called *suppress.* The appearance of the suppress code causes the processor to insert a flag in the input at that point and, in effect, to suppress the recording of output codes. While in suppress, the operations which determine horizontal and vertical spacing procede normally. During the trial set, computations based on the material set are possible. Changes in boundaries and other variables can be placed in effect after the trial set is concluded and before the reset begins. When the *unsuppress* code is encountered, the material following the flag placed by the suppress code is reset.

A three-line poem will be used to demonstrate a trial set. The trial begins with the assignment of the value zero to general variable one, starting suppress, and setting the left boundary to the value assigned to **gv1**; that is, to zero. Each line of the poem is terminated by a new line code; but first the length of the line is compared to the value of **gv1**. If line length is greater than **gv1**, **gv1** is assigned a value equal to line length.

At the end of the third line, **gv1** holds a value equal to the length of the longest line. The difference between measure and **gv1** is divided by two. At this point the unsuppress code signals the resetting operation. The first code following the flag placed by the suppress code sets the left boundary to the value assigned to **gv1**. The result is that the longest line is centered and the other lines are aligned with it. The input text follows:

```
[gv1,0;su;lb,gv1]
Ommnigenumque deum monstra et latrator Anubis[x1]
Contra Neptunum et Venerem contraque Minervam[x1]
Tela tenant.[x2]

Where: [sy,x1[[gt,cx,gv1[[gv1,cx]];nl]]]
       [sy,x2[[x1;df,gv1,rb,gv1;qo,gv1,gv1,2;us]]]
```

Execution of the example results in the following display of the poem:

Ommnigenumque deum monstra et latrator Anubis
Contra Neptunum et Venerem contraque Minervam
Tela tenant.

The suppress code, if stated with a non-zero parameter, provides for saving the trial set material toward the possibility that it will be acceptable. At the end of the trial, the unsuppress is written with a non-zero parameter to indicate that the material stands or without a parameter to indicate that the trial set material is to be discarded and the text following the flag placed by the suppress code is to be reset. The unsuppress codes with and without parameters will be written within conditional procedures devised to determine whether the trial set material is to be allowed to stand or is to be reset.

Save strings

The first and last entries on dictionary pages are usually printed prominently at the top of the page. The ability to define and store (and subsequently retrieve) strings of characters as the manuscript is being composed is provided by the save and execute string codes. Two such strings may be saved; they are identified by the numbers one and two. In effect, the parameter, one or two, is the name of the string being defined; the mark end serves to terminate both save strings.

The codes to note the extent of the input data to be saved are *mark begin* and *mark end.* The material so marked is then stored in addition to being composed —or executed if a portion of the marked material is code. The saved material is recalled subsequently for composition by the *execute saved string* code. The three codes are:

Mark begin:—	[**mb**,parameter]
Mark end:—	[**me**]
Execute saved string:—	[**es**,parameter]

When the saved string is called for execution, the font and size may, of course, be different.

This facility was used in the controls for composition of the PAGE–1 dictionary, Appendix A. The synonym names as they were used in manuscript are:

```
[v3]bb[v4]bottom boundary[v5]typo...
```

in which the final part of **[v3]** and the synonyms **[v4]** and **[v5]** are defined as follows:

```
[sy,v3[[...;lb,0;ps,10;bl,12;tf,6;mb,1]]]
[sy,v4[[me;ps,8;bl,10;tf,24] — [tf,14;mb,2]]]
[sy,v5[[me;tf,24] — ]]
```

Each time a new entry is composed, the code name and the English name are stored in save strings one and two respectively. When the bottom boundary of column two is encountered, the foregoing data for the last entry on the page reside in these two strings. A portion of the bottom boundary procedure, that which sets the right hand entry at the top of the page, is shown below:

```
...;ps,10;tf,14;es,2;tf,24] — [ps,12;tf,6;es,1;dn,11;...
```

Baseline displacement

The facility to displace the VideoComp baseline from its nominal position was used in the composition of the chapter headings of this book. *Baseline up* and *baseline down* statements are issued with a parameter to express the desired displacement in 1/32nd point increments.

Baseline up:—	[**up**,parameter]
Baseline down:—	[**dw**,parameter]

Beam displacement statements express absolute displacement from nominal or zero. Succeeding beam displacement statements also express displacement from zero; the excursions are not cumulative. The sequence **[...;up,32;...;up,64;...;up,0;...]** would result in a displacement of one point followed by a displacement of two points and finally a return to the nominal position. When these statements are used, the return to nominal baseline position may be by either baseline up or baseline down with a parameter of zero.

Ignore and consider

Used with conditional statements, the *ignore* and *consider* codes allow the composition of two (or more) different versions of a manuscript after but one data entry. From the moment the ignore code appears in input, the processor scans the input stream but takes no other action until the consider code appears. Composition is then resumed; the effect is to have omitted everything between the two codes.

Such things as teacher's and student's versions of the same text book, in which the teacher's book has answers but the student's does not, might be handled in the following way. The entire manuscript, including the answers, is typed once with the answers preceded by a conditional statement, which includes the ignore code, and followed by the consider code.

1. What is the relationship between the circumference and the diameter of a circle? [**x1**]Answer: π. [**x2**]

Where: **[sy,x1[[eq,gv1,0[[ig]];fl,.]]]** and
[sy,x2[[cn;nl]]]

If the sample is processed first with **gv1** equal to zero and then to one, the results are as follows:

[gv1,0] yields:

1. What is the relationship between the circumference and the diameter of a circle?

[gv1,1] yields:

1. What is the relationship between the circumference and the diameter of a circle?.................................. Answer: π.

Execute and fill to measure

These two codes, in a sense, direct the processor to generate material. Both are useful in generating rules; that is, lines. They serve other ends as well.

The *execute* code gives the capability to "execute" a string of control codes (or to set a passage of text) a specified number of times. The statement takes the form:

[**ex**,parameter[[*codes to be executed*]]]
[**ex**,parameter[*text to be set*]]
[**ex**,parameter[*text* [*code*]]]

in which the number of executions desired is stated in the parameter. The processor assigns the value of the stated parameter to **gv0** and subsequently decrements **gv0** one count with each execution. After execution has started, the number of remaining executions may be altered by assigning a new value to **gv0**.

The material supplied in the execute string may be explicitly stated control statements, a synonym name, a format label, text, or any combination thereof. For example, a vertical rule 624 points long will be generated by the following string of statements:

...;ps,16;bl,16;tf,4;ex,39[|[nl]];...

A statement of this sort was used to generate the vertical rule portion of the boarder around the pages of Appendix C. In the generation of vertical rules, body lead must equal point size to avoid gaps in the rule.

Fill to measure provides the facility to "leader out" a line with one or two text characters or, if there is no parameter, with white space; that is, to "flush" two components of a line to opposite ends of the line. The code may be used in any of the following forms:

[fl]
[fl,text character**]**
[fl,text character,text character**]**

The fill to measure code is placed in the text stream in the position the leader is to occupy; such a line must be terminated by a forced line break. The running head of Chapter Three which was accomplished by "flushing" the page number left and the chapter title right with a down zero between, can be specified as well by:

...;rm,gv199,pn,2;
eq,gv199,0[STRATEGY FOR THE AMERICAS**[fl;dc,pn]];**
eq,gv199,1[[dc,pn;fl]BACKGROUND FOR DECISION**];**
dn,24;...

Compare these statements with those on page 22.

If fill-to-measure is used with a single parameter, that character is repeated by the processor to form a character pair for the fill-to-measure action, or a character pair may be input as two different characters. The characters are placed on the line starting at a point which is a multiple of their combined widths away from the right boundary. This placement of the leader characters provides for individual fill characters to align vertically when fill-to-measure is used to leader succeeding lines.

If the line to be leadered starts with text, there may be space between the text and the first leader character. This space results from the relationship between the end point of the opening text and the start point of the leader as determined by the method just described. If the line to be leadered ends with text, there may be space between the last leader character and the start of the closing text.

Consider the following input stream:

[nl,1]Leading text**[fl,.,__;nl;fl,.,__;]**
closing text**[nl;fl,.,__;nl]**
Leading**[fl,.,__]**closing**[nl,1]**

Execution of this sample gives the following display:

```
Leading text  .__.__.__.__.__.__.__.__.__.__.__.__.__
  .__.__.__.__.__.__.__.__.__.__.__.__.__closing text
  .__.__.__.__.__.__.__.__.__.__.__.__.__.__.__.__.__
Leading.__.__.__.__.__.__.__.__.__.__.__.__.__   closing
```

A horizontal rule is generated by a code string of the following form:

[ps,8;tf,6;fl,—;nl]

Execution of this code string gives the following result:

```
————————————————————————————————————————
```

The point size chosen must be an even numbered submultiple of line length or the rule will be short at the left boundary. If this circumstance exists, it may be overcome by setting two em dashes as text characters followed by a down zero to terminate the line and then the fill to measure statement.

Manuscript identification

In general a job is composed once for proof and one or more times after that to accommodate changes and to eliminate errors. The system provides for selective recomposition of major portions of the job, called *logical units*, in order that the entire job need not be recomposed for each error correction cycle.

A logical unit may be as small or as large as the user wants it to be. The only restriction is that each logical unit start a new print region. The start of a logical unit is marked by the codes:

[rf,parameter;**pg]** or
[rf,parameter;**x1]**

in which **x1** is interpreted to be any legal synonym name or format label and further that the first language code in **x1** is **pg**. Parameter must be a decimal number in the range one through 4095 inclusive. The logical units that comprise a job must be numbered sequentially but not necessarily consecutively. Logical units may be entered on disc storage in any order; the processor will put them in the specified sequence at composition time.

Each logical unit may be subdivided into keyboard *takes* for ease in preparation of input material. The system provides for fifteen keyboard takes for each logical unit. Each manuscript take starts with a single character, which identifies the code produced by the input device, followed immediately by the *start take* statement which takes the following form:

[st,JOB,MRP,TK,B,T,XX]

In which:

JOB Job number, a decimal number in the range 1 to 999 inclusive, which identifies the composition job to which the following units belong.

MRP Manuscript reference point, a decimal number in the range 1 to 4095 inclusive, which identifies the logical unit to which the first of the units that follow belongs. (A keyboard take may comprise part of a logical unit, an entire logical unit, or several logical units according to the way the manuscript was divided for keyboarding.)

TK Take number, a decimal number in the range 1 through 15 inclusive, which identifies the take number within the logical unit.

B Job type; in Page–1, this character is always B.

T Take type, a single alphabetic character which may be **t** if the take is original manuscript or **a** if the take is alteration or correction material.

XX Two alphanumeric characters at the user's disposal; it is often used to identify the originator.

The start take statement must appear between its own left and right delimiters; the semi-colon may not be used to separate the last parameter of a start take statement from the PAGE–1 code which follows. The usual beginning sequence for a keyboard take is:

[st,400,1,1,b,t,jp]**[rf**,1;**pg]** or
[st,400,1,2,b,t,jp]text or control

preceded by the keyboard identifier. The first "header" above introduces the first take of the first logical unit of job 400 and it is followed by the manuscript reference point. The second introduces the second take of the same logical unit; it can start with text or control as the case may be.

PART III

ALGORITHMS FOR THE GENERATION OF PAGES

Part III takes up the development of pagination algorithms. An algorithm in the mathematical or computing sense is simply a set of rules by which a given problem is solved. In the PAGE–1 context, an algorithm—called a *Format Statement*—comprises a string of PAGE–1 words, or instruction codes, which will direct the PAGE–1 processor in the production of the desired VideoComp operating tape; that is, one which will produce pages as described by the style specification.

Chapter Five presents a very simple algorithm to generate numbered pages. This illustrates the function of the *when bottom* boundary procedure. An *at bottom* boundary procedure is then added to this simple "galley" algorithm to produce a new algorithm capable of typesetting a simple, single-column, book. Chapter Five ends with an algorithm for two-column text.

Chapter Six starts with the format statement for single-column text and introduces subsidiary format statements required in the typesetting of a straightforeward, non-fiction book. Such format statements are called subsidiary since they operate within the framework of the pagination algorithm to change font and size, provide vertical white space, or provide indentions as required by the style specification.

Chapter Seven considers the typographic requirements posed by footnotes. Format statements for one commonly used footnote treatment are developed and the single-column pagination algorithm is modified to accommodate them. An alternate treatment of footnote carryover is introduced and used as the basis for two-column footnote format statements; the two-column pagination algorithm is modified to accommodate them.

Chapter Eight introduces the PAGE–1 method of allocating white space during page composition for subsequent insertion of illustrations. Format statements for allocating white space in one and two-column text are developed and the algorithms of Chapter Seven are further modified to include this capability.

5

PAGINATION ALGORITHMS

THE STYLE SPECIFICATION of Figure 5.1 describes a text page of 38 lines and a page number which is centered at the top. The expression "type page" refers to the area in which type is set; it includes the body text and the page number (or page number and running head). The pages described are in effect simple galleys; the material which appears in input is composed such that a VideoComp will produce numbered pages of constant column width and depth in a single typeface and size.

The page described by the style specification is shown diagramatically in Figure 5.2. It comprises two blocks—block one for text and block two for the page number.

The opening page

Consider first the codes required to initiate the first page; the block origins are stated first.

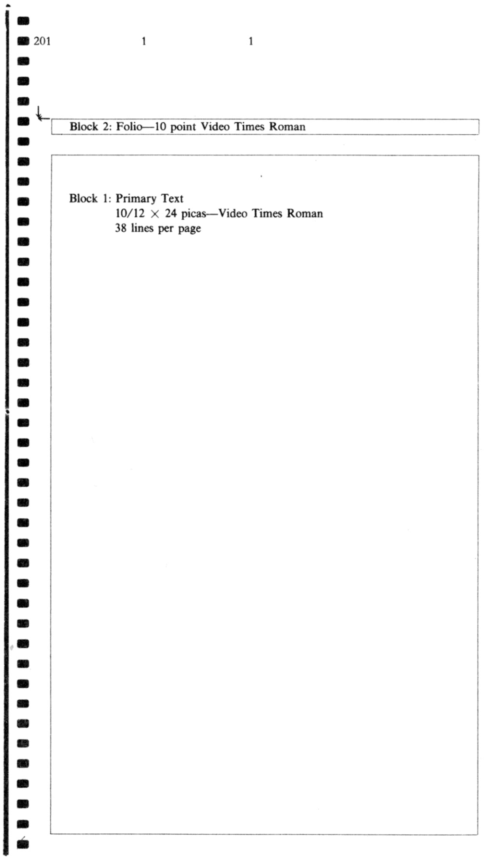
bx,1,12;by,1,24;bx,2,12;by,2,0;

The X-coordinates of both blocks are set to twelve; that is, both blocks are to be positioned one pica rightward from the inside edge of the sprocket holes on the left side of the final film. The Y-coordinate of the folio block—block two—is to be positioned level with the print region reference point. The Y-coordinate of the text block is to be positioned 24 points below that of the folio block. This satisfies the style specification that 14 points of white space be provided below the folio which is to be set in ten point type.

Figure 5.1
Style specification

Type page	24 × 40 picas—38 lines per page, 1 em paragraph indention
Typeface	Video Times Roman
Type size	10/12
Folio	10 point Video Times Roman, centered at the top of the page, 14 points below

Figure 5.2
Sample problem page


201
1
1
Block 2: Folio—10 point Video Times Roman
Block 1: Primary Text
10/12 × 24 picas—Video Times Roman
38 lines per page

Text block attributes

The attributes of the text block will be established first. The specification states that the measure—line length or the width of the printed page—is to be 24 picas which is accordingly stated as left boundary zero and right boundary 288 points. The top boundary is specified as zero and the bottom boundary equal to the depth required by 38 twelve point lines—456 points. The text block boundaries are stated following the *start block one* command.

```
sb,1;lb,0;tb,0;rb,288;bb,456;
```

The typeface, size, and leading are stated next followed by the justification mode.

```
ps,10;bl,12;tf,4;ju;
```

Video Times Roman, Size Range II (which includes point size ten), is typeface four.

The folio block

At this point the controls for typesetting in the text block are complete; controls to provide for the folio will now be considered. The text block is the primary block by virtue of having been started first. The text block is interrupted, which causes the processor to store all of the block one attributes—that is, the primary block attributes—for subsequent recall and allows definition of the interrupting block attributes. The folio block, in this case, becomes the interrupting block. The boundary and typeface statements are the same as those for the primary block but the justification mode is *center horizontally.*

```
ib;sb,2;lb,0;tb,0;rb,288;
ps,10;bl,12;tf,4;ce;
```

The controls for typesetting in the folio block are now complete. Next the page number will be typeset and the page number line terminated.

```
dc,pn;nl;cb;
```

The *continue block* command causes the attributes of the folio block to be stored and the attributes of the primary block to be recalled.

Summary of events

If the string of codes presented to this point is executed by the processor, the following summary of conditions will obtain:

- Two blocks together with their origins and attributes will be in existance. Block one is the primary or text block and block two is the interrupting or the folio block.
- The page number will have been set centered horizontally in block two; that is, at the top of the page.
- The primary block will be in force and the processor will be prepared to set text.

The processor will typeset material which appears in input— line by line downward in block one until the bottom boundary is encountered. A bottom boundary string which will turn the page is required.

Turning the page

The processor is setting text in block one when the bottom boundary is encountered and the *when bottom* boundary procedure is called for execution. The procedure begins with the command to start a new print region and the information that the continuation of the input material is to be typeset in block one. Next the page number is incremented one count. The text block is interrupted; its attributes and the continuing input point are stored by the processor. The folio block is started and the attributes of the interrupting block are called into force. In this case there is no other interrupting block involved; the *interrupt block code* will call block two and its attributes into force. The *start block* is stated explicitly here in the interest of uniformity and completeness. When footnotes are introduced, the explicit *start block* statement will be necessary since the footnote block will also be an interrupting block. The page number is typeset. Finally the page number line is terminated and the text or primary block is resumed.

```
pg;sb,1;ad,pn,pn,1;
ib;sb,2;dc,pn;nl;cb
```

The code string just developed is shown as a synonym in Figure 5.3.

Summary of events

If the text block bottom boundary is encountered and the *when bottom* procedure just developed is executed by the processor, the following summary of conditions will obtain:

- Two blocks together with their origins and attributes will be in existance.
- The chapter opening page—with the number centered at the top—will have been completed.
- The second page will have been started: the page number will have been incremented and typeset centered horizontally in block two.
- The primary block will be in force and the processor will be prepared to continue setting input text from the point at which it was interrupted.

Redundant codes

A number of redundant codes have been included in the format statement. This was done in the interest of presenting a systematic approach to the development of format statements. With a firm and straightforward method in mind, one is less likely to make errors.

As we have seen, initial conditions when the processor is first started provide for left and top boundaries to be zero; these two boundaries need not be stated since zero is the desired value in this case. The justify mode is also an initial condition. Further, at the first *interrupt block* command after the program is started, the attributes of the primary block are assigned as initial values for the interrupting block. Only those attributes of the interrupting block which are different from those of the primary block need be stated. In this case, only the justification mode is different. Finally, when the *new page* command is encountered, the program assumes that the continuing input material is to be typeset in the same-numbered block in the new print region.

Manuscript markup

If the synonym statement of Figure 5.3 were typed into input ahead of manuscript data, the only manuscript markup necessary consists of the code to initialize the page number counter and the label itself: **[pn,1;T1]**. Beyond these codes at the beginning of the input—following the *start take* statement and *logical unit* declaration—any desired changes in typeface, size, or boundaries, or any other desired change may be stated explicitly on the manuscript. The synonym statement will set numbered galleys 40 picas deep as long as the input lasts.

Figure 5.3
Initiation of blocks and pagination procedure for simple galley—T1

Block origins	[bx,1,12;by,1,24;bx,2,12;by,2,0;
Start text block	sb,1;
Text block attributes	
boundaries	lb,0;tb,0;rb,288;bb,456;
font	ps,10;bl,12;tf,4;
justification mode	ju;
Start folio block	ib;sb,2;
Folio block attributes	
boundaries	lb,0;tb,0;rb,288;
font	ps,10;bl,12;tf,4;
justification mode	ce;
Typeset page number	dc,pn;nl;
Resume text block	cb;
Pagination procedure	wb[[
New page	pg;sb,1;
Increment page number	ad,pn,pn,1;
Enter folio block	ib;sb,2;
Typeset page number	dc,pn;
Terminate line	nl;
Resume text	cb]]]

Figure 5.4
Style specification

Type page	25 × 76 picas—75 lines per page, 1 em paragraph indention
Typeface	Video Times Roman
Type size	10/12
Folio	10 point Video Times Bold, flushed right at the bottom of the page, two points above

Exercise: Write a synonym statement to specify the typesetting of the galley described in Figure 5.4; call it **T2**. Video Times Bold, Size Range II, is typeface six.

SINGLE COLUMN TEXT

Among the simplest books to be found is the Tarzan series. In general, books of this kind are typeset single column in a single typeface and size, they have no sub-heads, and no other typographic variation than the chapter head. Such a book might be described by the style specification of Figure 5.5.

The simple book

The first page of each chapter has the page number centered at the bottom of the page and a simple Roman numeral for a chapter head. Subsequent pages in each chapter have the page number centered at the top of the page. The chapter opening page and a continuation page are shown diagramatically in Figure 5.6.

Figure 5.5
Style specification

Type page	24 × 40 picas—38 lines per page, 2 em paragraph indention
Typeface	Video Times Roman
Type size	10/12
Folio	10 point Video Times Bold, centered, 14 points below
Drop folio	Chapter opening, 10 point Video Times Bold, centered, 2 points above
Chapter head	24 point Video Times Roman, Roman numeral, 6 picas above, 2 picas below

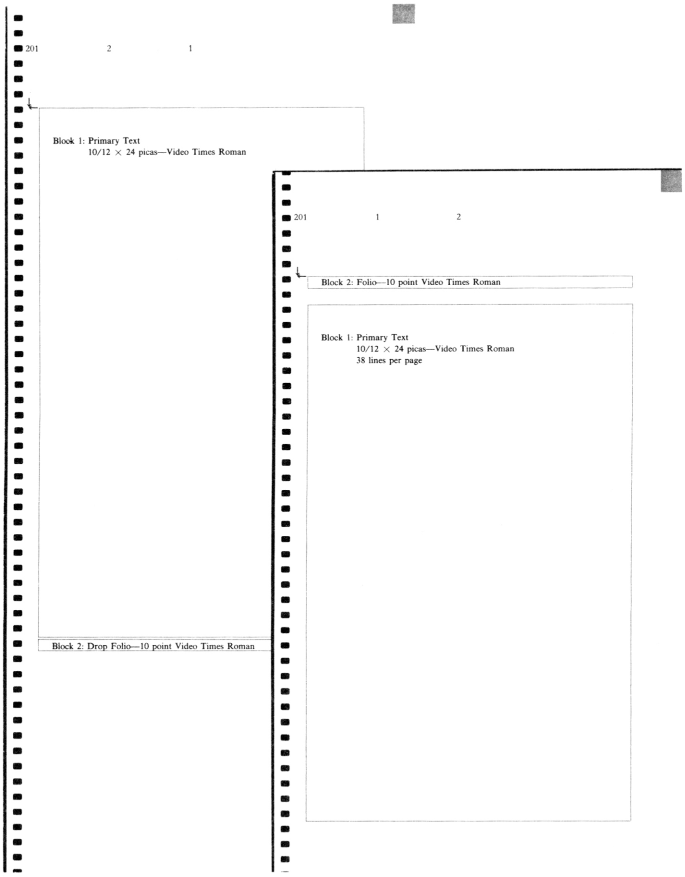
The pagination algorithm

The algorithm to be developed will open by describing a chapter opening page quite similar to **T2**. Two bottom boundary procedures will be specified: a *when bottom* boundary procedure—the same one used in **T1** —to close continuation pages and an *at bottom* boundary procedure to close the chapter opening page and to respecify block origins for the continuation pages.

The opening page

Consider first the portion of the format statement which describes the chapter opening page; it is shown in Figure 5.7. The origin of the text block is at the top of the page (level with the print region reference) and the origin of the folio block is 470 points below in order that the ten point folio will be at the bottom of the forty pica (480 point) type page. The bottom boundary of the text block is established to be 468 points so that there will be a minimum of two points above the drop folio.

Figure 5.6
Sample problem page


201
2
1
Block 1: Primary Text
10/12 × 24 picas—Video Times Roman
Block 2: Drop Folio—10 point Video Times Roman
201
1
2
Block 2: Folio—10 point Video Times Roman
Block 1: Primary Text
10/12 × 24 picas—Video Times Roman
38 lines per page

The bottom boundary procedures

The *when bottom* procedure, at the bottom of Figure 5.7, is the same as that for the simple galley case. It closes the page just completed, starts a new page and increments the page number counter, enters the folio block and sets the page number, and finally resumes text. The *at bottom* procedure includes, in addition to the entire *when bottom* procedure, the codes which change the Y-coordinate of the folio block to the top of the page and the Y-coordinate of the text block to a position 24 points below. This text block origin and the ten point folio provides the fourteen points below the folio required by the style specification.

Figure 5.7
Initiation of blocks for opening page and pagination procedures for a simple book—A3

Block origins	[bx,1,12;by,1,0;bx,2,12;by,2,470;
Increment page number	ad,pn,pn,1;
Start text block	sb,1;
Text block attributes	
boundaries	lb,0;tb,0;rb,288;bb,468;
font	ps,10;bl,12;tf,4;
justification mode	ju;
Start folio block	ib;sb,2;
Folio block attributes	
boundaries	lb,0;tb,0;rb,288;
font	ps,10;bl,12;tf,6;
justification mode	ce;
Typeset page number	dc,pn;nl;
Prepare to set text	cb;
At bottom procedure	ab[[
New page	pg;sb,1;
Change bottom boundary	bb,456;
Increment page counter	ad,pn,pn,1;
Enter folio block	ib;sb,2;
Typeset page number	dc,pn;
Terminate line	nl;
Resume text	cb;
Change block origins	by,1,24;by,2,0]];
When bottom procedure	wb[[
New page	pg;sb,1;
Increment page counter	ad,pn,pn,1;
Enter folio block	ib;sb,2;
Typeset page number	dc,pn;
Terminate line	nl;
Resume text	cb]]]

The bottom boundary of the text block is changed to 456 points to accommodate 38 twelve point lines. Note that the 456 point depth of the text block and the 24 point Y-coordinate of its origin meet the style specification requirement for a 40 pica (480) depth of the type page.

Vertical justification of the text blocks

The 38–12 point lines in the text block of a continuation page actually occupy only 454 points when the two points of lead on the last line are subtracted. This means that the type page is actually two points less than 40 picas deep. One way to take up this space and still maintain the fourteen points below the folio is to use the *justify vertically* code. The extra two points of space will be distributed between the lines in the text block. The depth of the final text block then, from the top ascender to the bottom descender, will be 456 points and the type page will have the depth required by the style specification.

The same situation exists in the chapter opening page. The chapter head occupies ten picas (six picas above and two picas below a 24 point Roman numeral) which leaves 348 points of depth for text on the chapter opening page. Twenty-nine lines occupy 346 points when the two-point lead of the bottom line is subtracted. The result without the *justify vertically* command is four points (rather than two) above the drop folio.

The *justify vertically* code has been added to the beginning of the bottom boundary procedures as they appear in Figure 5.9.

The chapter head

The format statement just developed and given the label **A3** will direct the processor in composing the pages described by the style specification of Figure 5.5 and diagrammed in Figure 5.6. The thing which remains is to develop format statements to direct the setting of the chapter head.

The style specification requires a six pica white space above and a two pica white space below the chapter head which is a simple Roman numeral set in 24 point Video Times Roman. The chapter opening page is shown diagramatically in Figure 5.8.

Assuming that the format statement **A3** has been executed by the processor, the program is ready to begin setting the opening page of a chapter. The vertical setting position is zero; that is, at the top edge of block one. The six pica sink above the chapter opening will be provided by a *non-justifiable down* command. The amount of the down must be 72 for the white space plus 24 for the point size of the character, or 96 points, because the baseline of the line following a down command is coincident with the vertical position resulting when the down command is executed.

The typeface must be changed to number 166 to select Video Times Roman, Size Range III (which includes point size 24), and the point size must be changed to 24. The two pica space below the chapter head can be provided at this point by specifying a body lead of 48 points. This completes the codes required to prepare for setting the chapter head.

Following the chapter head, a new line code followed by a change of font to 10/12 Video Times Roman, Size Range II, is all that is required to prepare for text. Both statements appear near the bottom of Figure 5.9.

The format statements of Figure 5.9 are one means of compliance with the postulated style specification.

Figure 5.8
Chapter heading

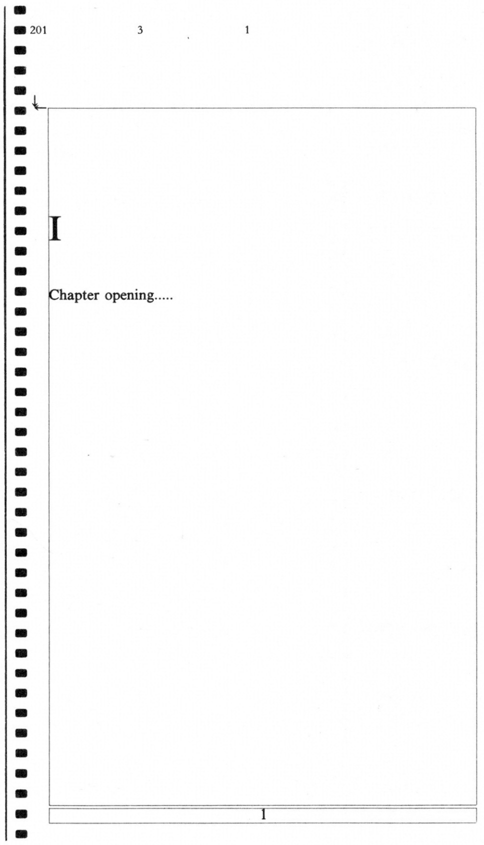


Figure 5.9
Format statements for single column book

```
A3≡ [bx,1,12;by,1,0;bx,2,12;by,2,470;ad,pn,pn,1;
        sb,1;lb,0;tb,0;rb,288;bb,468;ps,10;bl,12;tf,4;ju;ib;
        sb,2;lb,0;tb,0;rb,288;ps,10;bl,10;tf,6;ce;
        dc,pn;nl;cb;
        ab[[jv;pg;sb,1;bb,456;ad,pn,pn,1;
            ib;sb,2;dc,pn;nl;cb;by,1,24;by,2,0]];
        wb[[jv;pg;sb,1;ad,pn,pn,1;ib;sb,2;dc,pn;nl;cb]]]

A6 ≡ [dn,96;ps,24;bl,48;tf,166]

A8 ≡ [nl;ps,10;bl,12;tf,4]
```

They demonstrate a point that will recur many times: sets of format statements are required to specify the typesetting of a book and the statements in a set must be compatible with each other.

Another means of meeting the requirements of the style specification is to make the Y-coordinate of the block one origin for the opening page equal to 72 and to shorten the text block bottom boundary by 72 points. In this event the down command would be omitted from **A6**. It is also possible to make the initial font specification for block one (in A3 of Figure 5.9) 24/48 Video Times Roman. In this event **A6** is not needed at all and **A8** can be used to change the font to that required for text after the chapter head has been typeset.

TWO COLUMN TEXT

AN ALGORITHM FOR TWO COLUMN TEXT requires two text blocks and each must have its own bottom boundary procedure. The bottom boundary procedure in the left-hand text block will direct that typesetting proceed in the right-hand text block while the bottom boundary procedure in the right-hand text block will direct the page change, the following page folio, and the continuation of typesetting in the left-hand text block.

Figure 5.10
Style specification

Type page	31 × 48 picas: two 15 pica columns with 1 pica gutter—46 lines per column, 1 em paragraph indention
Text	10/12 × 15 picas, Video Times Roman
Folio	8 point Video Technica Bold, centered, 18 points above

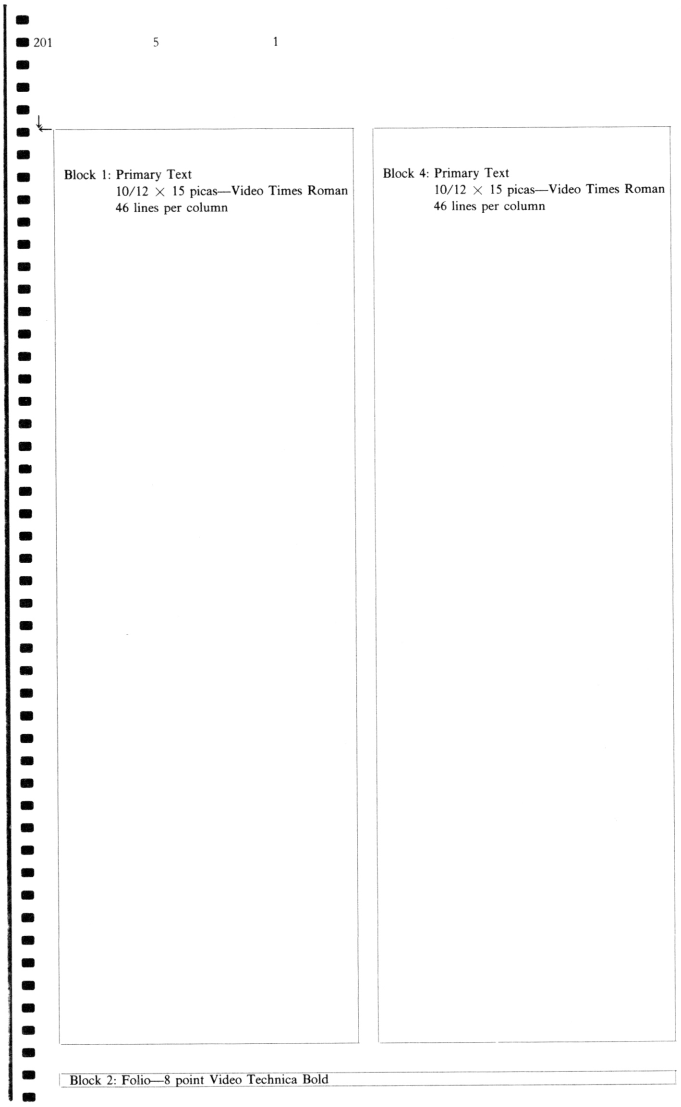
The pagination algorithm

The algorithm to be developed is based on the style specification given in Figure 5.10; the block map for the pages to be generated is shown in Figure 5.11. In this simple case, provision is not made for a full, two-column, chapter head; chapter heads are assumed to occupy only one column. The question of balancing the (partial) columns on the concluding page of the chapter—if this is desired—is left for consideration in a subsequent chapter.

The opening page

The opening portion of the format statement,

Figure 5.11
Simple two-column page


201
5
1
Block 1: Primary Text
10/12 × 15 picas—Video Times Roman
46 lines per column
Block 4: Primary Text
10/12 × 15 picas—Video Times Roman
46 lines per column
Block 2: Folio—8 point Video Technica Bold

shown in Figure 5.12, positions the blocks on the print region. The origins of the text blocks are at the top of the page and the origin of the folio block is 568 points below in order that the eight-point folio will be at the bottom of the 48 pica (576 points) type page. The X-coordinate of block four is 192 points rightward from that of block one to provide a one pica gutter between the fifteen pica columns. General variable one, initialized to zero, is a column indicator; its function is discussed subsequently.

Next the left-hand text block is started and its attributes are stated. The left and top boundaries are zero. The right boundary is 180 points corresponding to the 15 pica measure of the style specification. Forty-six text lines occupy 550 points (46 lines of ten point type and 45 occurrences of two point interline lead); this value is assigned to the bottom boundary. The 550 point bottom boundary of the text block and the 568 point Y-coordinate of the folio block origin provide 18 points between text and folio. The font, size, and justification mode for primary text conclude the definition of block one.

Figure 5.12
Initiation of blocks for two-column page and pagination procedure

```
Block origins               [bx,1,12;by,1,0;bx,2,12;
                             by,2,568;bx,4,204;by,4,0;
Reset column indicator      gv1,0;
Increment page counter      ad,pn,pn,1;
Start text block            sb,1;
Text block attributes
              boundaries        lb,0;tb,0;rb,180;bb,550;
                    font        ps,10;bl,12;tf,4;
      justification mode        ju;
Start folio block           ib;sb,2;
Folio block attributes
              boundaries        lb,0;tb,0;rb,372;
                    font        ps,8;bl,8;tf,8;
      justification mode        ce;
Typeset folio               dc,pn;nl;
Prepare to set text         cb;
Pagination procedure        wb[[
LH procedure                    eq,gv1,0[[
   vertical justification               jv;
         start RH column                sb,4;
  reset column indicator                gv1,2]];
RH procedure                    eq,gv1,1[[
   vertical justification               jv;
                new page                pg;
         start LH column                sb,1;
      bump page counter                 ad,pn,pn,1;
      enter folio block                 ib;sb,2;
   typeset page number                  dc,pn;nl;
             resume text                cb]];
Decrement column indicator      df,gv1,gv1,1]]]
```

After the text block attributes are stated, the text (the primary) block is interrupted and the folio (the interrupting) block is started. The boundaries—zero and 372 points for left and right boundaries, respectively—are set to correspond to the left edge of the left-hand text column and the right edge of the right-hand column. Video Technica Bold, Size Range II, is typeface eight. The folio is typeset next and then typesetting in the primary block is resumed. Finally, the pagination procedure completes the algorithm.

The bottom boundary procedure

Since in this case there is no difference between chapter opening and continuation pages, only a *when bottom* boundary procedure is required. The procedure will comprise actually two distinct procedures—one for each text column—and a column indicator which is used at each encounter of a text block bottom boundary to select the proper procedure for execution. Another common technique makes use of a *when bottom* procedure which defines an *at bottom* procedure.

General variable one will be used as the column indicator; it will be manipulated in the bottom boundary procedure such that when the bottom boundary of block one is encountered, **gv1** has the value zero and when the bottom boundary of block four is encountered, **gv1** has the value one. The appropriate procedures then are both included, as conditional procedures, in a single *when bottom* boundary procedure.

Consider the bottom boundary procedure of Figure 5.12. The value of **gv1** is first tested for zero and if the bottom boundary of block one has just been encountered, the value is zero. The conditional procedure calls for vertical justification of the current text block

Figure 5.13
Format statement for two-column text

```
A4 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,568;
     bx,4,204;by,4,0;gv1,0;ad,pn,pn,1;

     sb,1;lb,0;tb,0;rb,180;bb,550;ps,10;bl,12;tf,4;ju;ib;

     sb,2;lb,0;tb,0;rb,372;ps,8;bl,8;tf,8;ce;dc,pn;nl;cb;

     wb[[eq,gv1,0[[jv;sb,4;gv1,2]];
          eq,gv1,1[[jv;pg;sb,1;ad,pn,pn,1;ib;sb,2;dc,pn;nl;cb]];
          df,gv1,gv1,1]]]
```

(block one) followed by resumption of text in block four. The value of the column indicator is then set to two and the conditional procedure is closed. The value of **gv1** is then tested for one—its value is two and the second conditional statement is not executed—and finally the value of **gv1** is decremented one count (to one) and the execution of the *when bottom* boundary procedure is complete. The typesetting of text will proceed in block four.

When the bottom boundary of block four is encountered, the value of **gv1** is one. The first conditional is not executed but the second one is. Text block four is vertically justified, a new page is started, and text block one is established as the primary block. Next the page number counter is incremented, the folio block is entered and the folio is typeset, and finally the text block is resumed. The conditional procedure is closed. Then the value of **gv1** is decremented one count (to zero) and this execution of the *when bottom* boundary procedure is completed. The typesetting of text will procede in block one. The format statement for two-column text is shown in Figure 5.13.

Exercise: 1. Write a new pagination algorithm for **A4** using a *when bottom* procedure which defines an *at bottom* procedure; call it **T3**.

2. Many directories are typeset in a three or even a four column format. Write a pagination algorithm to specify the typesetting of the pages described in the specification of Figure 5.14. Use blocks one, two, and three as text blocks and four as the folio block. Video Times Roman, Size Range I, which includes point size seven, is typeface five. State the algorithm as **T4**.

Figure 5.14
Style specification

Type page	37½ × 48 picas: three 12 pica columns with nine point gutters—62 lines per column
Text	7/9 × 12 picas, Video Times Roman
Folio	10 point Video Times Bold, centered, 10 points above

Answers:—

```
[sy,T2[[bx,1,12;by,1,0;bx,2,12;by,2,902;
        sb,1;lb,0;tb,0;rb,300;bb,900;ps,10;bl,12;tf,4;ju;ib;
        sb,2;lb,0;tb,0;rb,300;ps,10;bl,12;tf,6;ce;dc,pn;nl;cb;
        wb[[pg;sb,1;ad,pn,pn,1;ib;sb,2;dc,pn;nl;cb]]]]]
[sy,T3[[bx,1,12;by,1,0;bx,2,12;by,2,568;bx,4,204;by,4,0;ad,pn,pn,1;
        sb,1;lb,0;tb,0;rb,180;bb,550;ps,10;bl,12;tf,4;ju;ib;
        sb,2;lb,0;tb,0;rb,372;ps,8;bl,8;tf,8;ce;dc,pn;nl;cb;
        wb[[jv;sb,4;ab[[jv;pg;sb,1;ad,pn,pn,1;ib;sb,2;dc,pn;nl;cb]]]]]]]
[sy,T4[[bx,1,12;by,1,0;bx,2,165;by,2,0;bx,3,318;by,3,0;bx,4,12;by,4,566;
        sb,1;lb,0;tb,0;rb,144;bb,558;ps,7;bl,9;tf,5;ju;ib;
        sb,2;lb,0;tb,0;rb,450;ps,10;bl,10;tf,4;ce;dc,pn;nl;cb;
        wb[[jv;eq,gv1,0[[sb,2;gv1,3]];eq,gv1,2[[sb,3]];
             eq,gv1,1[[pg;sb,1;ad,pn,pn,1;ib;dc,pn;nl;cb]];
           df,gv1,gv1,1]]]]]
```

6

SUBSIDIARY FORMAT STATEMENTS

THE FORMAT STATEMENT for single column text—developed in Chapter Five and shown in Figure 5.9—can be restated as shown in Figure 6.1. The only difference between the format statements in the two figures is in the statement of **dc,pn**; in Figure 5.9 it is explicit while in Figure 6.1 it is entered via the synonym **Y2**. In the case of Figure 6.1, the synonym **Y2** must be defined before the first call of the format statement **A2**.

The advantage of the format statement of Figure 6.1 is that the basic algorithm can be used to produce pages with a wide variety of folio treatments without re-keyboarding the basic statement, slightly modified each time. Different folio requirements can be included by writing and keyboarding the controls necessary to comply with each new book specification. Another possibility is that format statements may be written (and loaded on the system) for a variety of common folio treatments. Then each time preparations are made for typesetting a new book, the synonym **Y2** can be defined as the format label which corresponds to the folio treatment desired.

Figure 6.1
Format statement for single column text

```
A2 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,470;ad,pn,pn,1;
     sb,1;lb,0;tb,0;rb,288;bb,468;ps,10;bl,12;tf,4;ju;ib;
     sb,2;lb,0;tb,0;rb,288;ps,10;bl,10;tf,6;ce;Y2;nl;cb;
     ab[[jv;pg;sb,1;bb,456;ad,pn,pn,1;
         ib;Y2;nl;cb;by,1,24;by,2,0]];
     wb[[jv;pg;sb,1;ad,pn,pn,1;ib;Y2;nl;cb]]]

Y2 ≡ [dc,pn]
```

Flush folios outside

Figure 6.2 is a style specification for a book essentially similar to the book considered in Chapter Five; the only difference is that the folio consists of the page number flushed outside on chapter continuation pages. The format statement **A2** as it appears in Figure 6.1 will set the book if the folio controls are developed and given the synonym name **Y2**.

In general the right-hand pages of books carry odd numbers. The format statement to be developed will assume that this is so. The folio format statement—shown in Figure 6.3—begins by setting the justification mode to flush right on the assumption that the page is a right-hand page. Next, the current value of the page number counter is divided by two and the remainder which accrues is assigned to **gv199**. If the page number is odd, the remainder will be one and if the page number is even, the remainder will be zero.

Figure 6.2
Style specification

Type page	24 × 40 picas—38 lines per page, 1 em paragraph indention
Typeface	Video Times Roman
Type size	10/12
Folio	10 point Video Times Bold, flush outside, 14 points below
Drop folio	Chapter opening, 10 point Video Times Bold, centered, 2 points above
Chapter head	24 point Video Times Bold, Roman numeral, 6 picas above, 2 picas below

Figure 6.3
Format statement for folio—page number flush outside

Flush right	[qr;
Divide page number by two: store remainder	rm,gv199,pn,2;
Flush left if page number is even	eq,gv199,0[[ql]];
Typeset page number	dc,pn]

A single conditional statement appears next to change the justification mode to *flush-left* if the page number is even; that is, if the value of **gv199** is zero. If the value of **gv199** is one, the conditional statement is not executed and the *flush-right* justification mode initially set is allowed to stand. Finally, the code which causes the processor to typeset the page number concludes the statement. The folio statement of Figure 6.3 may be given the synonym name **Y2** and be keyboarded into input at the beginning of the job prior to the first call of the format statement **A2**.

The statement may be given a format label; for example, **G3**, and be loaded on the system. In this case, the synonym **Y2** must be equated with the format statement **G3** in the input prior to the first call of the format label **A2**; that is, **[sy,Y2[[G3]]]**.

Running heads with folio

A commonly used folio treatment is one in which page numbers are set flush outside and two different running heads are set flush inside. In some novels and many non-fiction books, the book title is used as the left-hand running head and the chapter title is used as the right-hand running head. In many scientific and technical books, the chapter title and the major chapter division subheads are used as left-hand and right-hand running heads respectively.

Figure 6.4
Format statement for folio—page number flush outside and two different running heads inside

Divide page number by two: store remainder	[rm,gv199,pn,2;
Left hand conditional	eq,gv199,0[[
typeset page number	dc,pn;
fill to measure	fl;
left hand running head	z1]];
Right hand conditional	eq,gv199,1[[
right hand running head	z2;
fill to measure	fl;
typeset page number	dc,pn]]]

The format statement shown in Figure 6.4 will provide the folio treatment just described. Since this treatment requires two different running heads, two conditional statements are necessary to provide both alternatives. As before, the page number counter is divided by two and the remainder is assigned to **gv199.**

If the remainder is zero, denoting a left-hand page, the page number is set first followed by the running head. The synonym name **Z1** is imbedded in the format statement to allow the user to insert the running head text when the statement is used. Use of the synonym in this way allows the text to be changed by redefining the synonym and calling the format statement which includes the synonym name. In the present case, the synonym is redefined at the beginning of each new chapter, before the format statement call. The page number and the running head are separated by a *fill-to-measure* code to cause them to be flushed to opposite ends of the line. The fill-to-measure action is initiated with the *new line* code which follows **Y2** in the pagination algorithm, **A2**.

If the remainder is one, denoting a right-hand page, the running head is set first followed by the page number. The synonym name **Z2** is imbedded in the format statement to allow the user to insert a different running head text for the right hand page. Again, the running head and the page number are separated by a *fill-to-measure* code.

Using the folio/running head format statement

The approach used here allows the folio/running head statements to be quite straight-forward. The folio block location, boundaries, and font are specified in the pagination algorithm. The folio format statements can, of course, be used with any other page generating algorithm which has been developed in the same general way as was **A2**.

In use, the text of both running heads must be specified appropriately. If the book title is to be used as the left-hand running head, the synonym **Z1** is defined to be the words of the book title—single line and appropriate shift for all caps or for small caps or upper and lower case as specified—at the beginning of the first take and before the first call of the format label **A2**. If the chapter title is to be used as the right-hand running head, the synonym **Z2** is defined to be the words of the chapter title—again single line and with appropriate shift—at the beginning of each new chapter before the format label **A2**.

The statements developed are shown in Figure 6.5.

These statements set the folio and the running heads in the same typeface and size specified in the format statement with which they are used subject only to the shift codes in the synonym definitions for the names **z1** and **z2**.

Exercise: Not infrequently, the style specification will call for the folio and running heads to be set in different fonts or in different sizes or both. A separate folio/running head format statement can be developed to meet this sort of requirement or the appropriate font codes can be included in the definitions of the synonyms **z1** and **z2**.

1. Write a folio/running head statement, similar to **G4**, to fulfill the following requirements; call the statement **U1**.

- Page numbers flush outside and two different running heads flush inside;
- Page numbers set in 10 point Video Times Bold and running heads in 8 point Video Times Roman.

2. Write a folio/running head statement to fulfill the following specification and call it **U2**.

- Page number flush outside on left-hand page;
- Chapter title and page number, separated by an em space, flush outside on the right-hand page.
- Assume the same font and size as specified in **A2**.

SECONDARY TEXT AND SIDE HEADS

THE FORMAT STATEMENTS developed in Chapter Five—with the addition of several new format statements to specify requirements for secondary text and for side heads—are sufficient to handle typesetting of a simple non-fiction book. Secondary text is used to draw attention to, or to set apart, extracts or quotes from another source. It is generally set in a smaller size in the same typeface used for body text; it may be indented at the left margin. See Figure 6.7. The remaining requirements for a more general, non-fiction book—footnotes and graphics—will be considered in the next chapters. The style specification for the book under consideration is shown in Figure 6.6.

Figure 6.5
Folio format statements

```
G3 ≡ [qr;rm,gv199,pn,2;eq,gv199,0[[ql]];dc,pn]

G4 ≡ [rm,gv199,pn,2;eq,gv199,0[[dc,pn;fl;z1]];
                        eq,gv199,1[[z2;fl;dc,pn]]]

G6 ≡ [ce;dc,pn]
```

Secondary text

If one assumes that extract material, or secondary text, is always preceded by and followed by primary text, the format statement to initiate secondary text needs only to:

- Issue a new line command and specify an additional two point sink,
- Change point size and body lead to 9/11, and
- Move the left boundary rightward 18 points.

The statement to resume primary text after secondary text must:

- Issue a new line command and specify an additional two point sink,
- Change point size and body lead to 10/12, and
- Move the left boundary back to zero.

Alternatively the *new line* and *down* commands could be replaced with *down* commands of 14 points and 13 points respectively. The relationships are shown graphically in Figure 6.7.

Unfortunately things are rarely so simple. If the line preceding the secondary text has just fit measure and justification has been effected, the horizontal setting position is zero and the processor is ready to start a new line. If the white space above the secondary text is specified by the "down-fourteen-points" method, the result will be sixteen points rather than four

Figure 6.6
Style specification

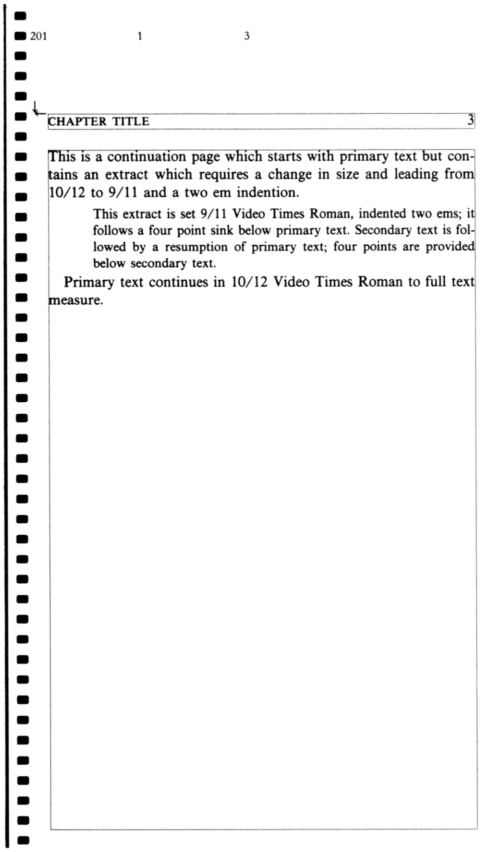
Type page	24 × 40 picas—38 lines per page, 1 em paragraph indention
Text	10/12 Video Times Roman
Extracts	9/11 Video Times Roman, indented 2 ems, 4 points above and below
Folio	10 point Video Times Bold, flush outside, 14 points below
Running head	10 point Video Times Bold small caps, flush inside, left—book title and right—chapter title
Drop folio	Chapter opening, 10 point Video Times Bold, centered, 2 points above
Chapter head	24 point Video Times Bold, Roman numeral, 6 picas above, 2 picas below
Side heads	
first	10/12 Video Times Bold, c/lc, flush left, 12 points above and 6 points below, turnovers flush left
second	10/12 Video Times Italic, c/lc, flush left, 10 points above and 2 points below, turnovers flush left

points. The "new-line and down-two-points" will give the proper result since a *new line* command is ignored if **cx** is zero.

This would seem a good reason for using the "new-line and down-two-points" approach, but this method is heir to possible trouble too. If the text preceding secondary text just completed a page, the bottom boundary procedure was executed and the format label introducing secondary text is the first thing the processor encounters on the new page. The result is a two point sink above the secondary text; a result which cannot be tolerated. Additionally there is the possibility, perhaps remote, that secondary text follows a first or second level side head without intervening primary text.

The question of how much to handle automatically and what situations to handle by explicit markup is a practical one. The situation in which secondary text follows a side head is immediately apparent in the manuscript and probably occurs rarely; it can be handled in markup. The other situations are much more likely to occur and cannot be anticipated in manuscript. Provision should therefore be made to handle them automatically.

Figure 6.7
Secondary text



201 1 3

CHAPTER TITLE 3

This is a continuation page which starts with primary text but contains an extract which requires a change in size and leading from 10/12 to 9/11 and a two em indention.

This extract is set 9/11 Video Times Roman, indented two ems; it follows a four point sink below primary text. Secondary text is followed by a resumption of primary text; four points are provided below secondary text.

Primary text continues in 10/12 Video Times Roman to full text measure.

The problem cases noted above can both be detected by testing the values of the vertical and horizontal setting positions for zero.

First: is the vertical setting position equal to zero? If it is, horizontal setting position may be zero or it may have some value less than line length. If **cx** is zero, the secondary text starts the text column and no space need be provided above it. If **cx** is not zero (but **cy** is zero), a partial line has been set—a widow, by the way—and the four points of space between that line and the secondary text must be provided.

Only the second half of the question need be explicitly stated in the format statement. The first half may be implied since the result is to provide no space. See the second line of Fgure 6.8.

Second: is the vertical setting position unequal to zero? If this is the case, the secondary text is not first on the page. The same two questions with regard to **cx** need to be asked. If **cx** is equal to zero, only two additional points of space need be provided. If **cx** is not zero, a fourteen point sink is required. See the first line of Figure 6.8.

The format statement to specify the typesetting of secondary text is concluded by the codes to change the point size and body lead to 9/11 and to move the left boundary rightward by eighteen points (two nine-point ems).

The format statement to specify the return to primary text—also shown in Figure 6.8 —is similar. The differences are a sink of 13 points, the change of point size and body lead to 10/12, and the return of the left boundary to zero.

Side heads

All of the spacing considerations of the secondary text treatment apply to the typesetting of side heads. For the space above the first side head, the sink re-

Figure 6.8
Format statements for secondary text

```
J1 ≡ [ne,cy,0[[eq,cx,0[[dn,2]];ne,cx,0[[dn,13]]]];
        eq,cy,0[[ne,cx,0[[dn,14]]]];ps,9;bl,11;lb,18]

J2 ≡ [ne,cy,0[[eq,cx,0[[dn,2]];ne,cx,0[[dn,14]]]];
        eq,cy,0[[ne,cx,0[[dn,14]]]];ps,10;bl,12;lb,0]
```

quired to provide twelve points above the side head are twenty-two and ten corresponding to **cx** equal-to or not-equal-to zero respectively. The statement is concluded with the codes to change the typeface to Video Times Bold and the justification mode to flush left. See Figure 6.9.

The processor will break long side heads and flush turnovers left as the style specification requires. Most book designers want long side heads to be broken into lines of approximately equal length. In this case, the format statement need not change the justification mode to *flush left* but the markup person must estimate the length of all side heads and break them appropriately with *new line* codes.

The format statement to initiate the return to primary text need not ask all of the earlier questions in order to determine spacing. A simple "new-line and down four-points" will provide the required six points below the first level side head. This format statement is heir to trouble: if its execution causes the close of a page, that page is apparently short by up to six points; if it opens a new page, there is an undesirable four points before the start of text. These, not-unlikely, events should be handled in later alteration; a result of the fact that the solution to this circumstance is always drastic. It requires human decision—often rewrite of the text material.

Figure 6.9
Format statements for side heads

```
H1 ≡ [ne,cy,0[[eq,cx,0[[dn,10]];ne,cx,0[[dn,22]]]];
        eq,cy,0[[ne,cx,0[[dn,22]]]];tf,6;ql]
H2 ≡ [nl;dn,4;tf,4;ju];

H3 ≡ [ne,cy,0[[eq,cx,0[[dn,8]];ne,cx,0[[dn,20]]]];
        eq,cy,0[[ne,cx,0[[dn,20]]]];tf,14;ql]
H4 ≡ [nl;tf,4;ju]
```

The format statements to provide for the second level side head are also shown in Figure 6.9; they are similar to those for the first level side head. Typeface fourteen is the Size Range II Video Times Italic font.

The format statements developed in this section are termed subsidiary since they operate essentially independently from the pagination algorithm. They change font and size, and provide spacing and indention; but they do not interact directly with the format statement which generates pages.

Exercise: The book under consideration may require two chapter heads, as shown in Figure 6.10. Write format statements to direct the setting of the two chapter heads and to open chapter text.

Figure 6.10
Style specification

⋮

Chapter heads	
number	24 point Video Times Bold, 7 pica indention, 2½ pica sink from top of type page, 2 picas below, arabic numeral
title	18 point Video Times Bold, 7 pica indention, 2 picas above and below, c/lc, 1½ picas between lines, break title into approximately equal lines
Initials	None, chapter opening indented 7 picas

⋮

Answers:—

```
[sy,U1[[rm,gv199,pn,2;
        eq,gv199,0[[ps,10;tf,6;dc,pn;fl;ps,8;tf,4;z1]];
        eq,gv199,1[[ps,8;tf,4;z2;fl;ps,10;tf,6;dc,pn]]]]]
[sy,U2[[rm,gv199,pn,2;
        eq,gv199,0[[ql;dc,pn]];
        eq,gv199,1[[qr;z2;hn,1;dc,pn]]]]]
[sy,v1[[dn,54;ps,24;bl,24;tf,166;lb,84;ql]]]
[sy,v2[[dn,42;ps,18;bl,36;tf,166;lb,84;ql]]]
[sy,v3[[dn,34;ps,10;bl,12;tf,4;lb,84;ju;ar[[lb,0]]]]]
```

7

PROCEDURES TO SET AND TO POSITION FOOTNOTES

FOOTNOTES ARE COMMONLY SET at the bottom of the page on which they are cited and in a smaller point size in the same typeface used for primary text. The footnote is usually cited in the text by a superior arabic numeral and the footnote itself is introduced by the same number which may be a superior number or may be in the same point size used for the footnote text. The first line of a footnote may or may not be indented.

If a footnote will not entirely fit on the page of its citation, it may be continued at the bottom of the succeeding page ahead of any footnotes which appear on that page. Some publishers like to have a full column rule (or a short rule starting at the left edge of the type area) above a footnote continuation.

Most of the detail of footnote treatment is governed by common practice or by understanding between publisher and typesetter. The style specification of Figure 7.1 is representative.

Figure 7.1
Style specification

	.
	.
	.
Footnotes	8/10 Video Times Roman, cited by consecutive, superscript, Arabic numerals, first footnote in each chapter is numbered 1, 2 points above
	.
	.
	.
	.

Page–1 approach to setting footnotes

Two format statements are required to direct the setting of footnotes: one to introduce the footnote and one to resume primary text after the conclusion of the footnote. During input preparation, the start footnote format label will be marked on the manuscript and typed into input at the citation point in primary text. The text of the footnote is then typed into input and is followed by the end footnote format label. Primary text is then resumed at the point following the interruption.

Footnote citations in text and with the footnotes will be set using the *set-decimal-number* code. The footnote will be positioned by setting the footnote in a separate block. After the footnote has been set—and its depth is known—the footnote block position will be computed and fixed. See Figure 7.2.

Format statements for footnotes

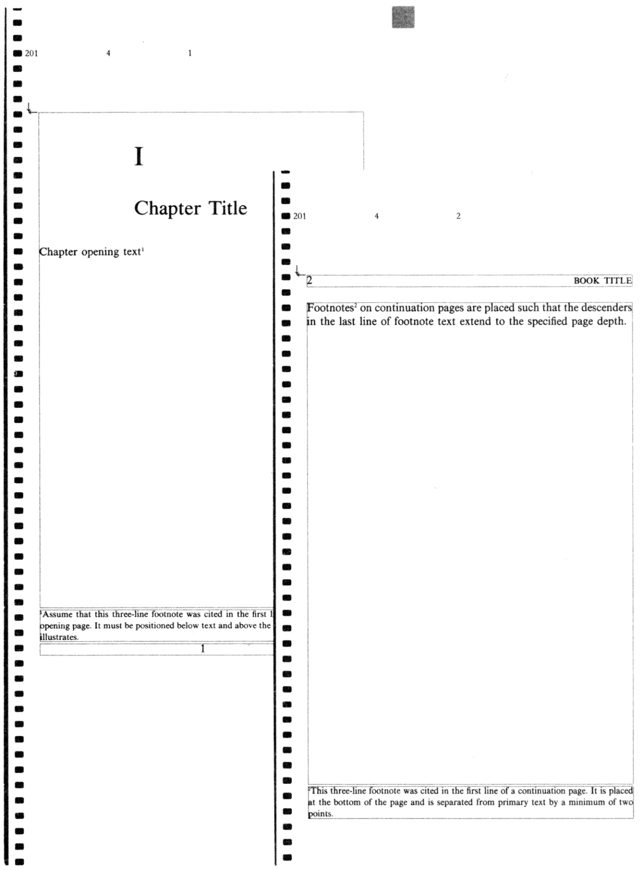
The footnote format statements will be developed in three steps:

- First—assuming that the footnote will fit on the page in which it is cited, statements to initiate footnote setting and to resume primary text will be developed.
- Second—simple carryover, assuming no carryover from the preceding page, will be considered and the footnote format statements will be modified.
- Third—carryover from the preceding page will be incorporated.

Modification to the pagination algorithm

The format statement for single column primary text, as it appeared in Chapter Five, modified to provide for elementary footnotes (that is, footnotes which fit on the page of their citation), is shown in Figure 7.3 with the additions set in boldface type. The footnotes will be set in block three; the x-coordinate is therefore set to twelve to be compatible with the primary text and folio blocks. The end footnote format statement will need a reference from which to compute the location of the footnote block and the depth remaining in the text block. The *page depth* global variable will provide the former reference and **gv89** will provide the latter. The global variable, **pd**, is initialized to 468 points for the chapter opening page; this is the maximum depth to which footnotes, or text, may extend on the chapter opening page. The drop folio is below text at the bottom of the page. On continuation pages, maximum page depth is 40 picas; the *page depth* global variable is, therefore, changed to 480 points in the *at bottom* procedure. General variable 89 is initialized to 468 points (the depth of text on the chapter opening page) and changed to 456 points in the *at bottom* procedure. The footnote citation is initialized to one.

Figure 7.2
Sample footnotes



201 4 1

I

Chapter Title

Chapter opening text[1]

[1]Assume that this three-line footnote was cited in the first l
opening page. It must be positioned below text and above the
illustrates.

1

201 4 2

2 BOOK TITLE

Footnotes[2] on continuation pages are placed such that the descenders in the last line of footnote text extend to the specified page depth.

[2]This three-line footnote was cited in the first line of a continuation page. It is placed at the bottom of the page and is separated from primary text by a minimum of two points.

The footnote block must be established as the interrupting block immediately after the folio/running head is set. This action simplifies the start footnote format statement: an *interrupt block* command is all that is required to enter the footnote block. In addition to the *start block* three command, the justification mode for footnotes is established. (The font and size are stated in the start footnote statement and need not be included here.) The codes to start the footnote block appear in three places: after the folio block in the part of the format statement which sets the chapter opening page and in both bottom boundary procedures.

As each new page is initiated, through execution of the bottom boundary procedure, the interrupting block in force is the footnote block. The commands to *start block* two and to change to the proper font and size to set the folio and running head must be added to both bottom boundary procedures.

Finally, if a footnote is set on any page, the end footnote format statement will shorten the text block bottom boundary by the amount of the combined depth of the footnote and the space specified between text and footnotes. The *when bottom* procedure must reset the bottom boundary of the text block on each new page to the standard 456 points.

Start footnote format statement

The format statement to initiate setting an elementary footnote may now be considered; the statement is shown in Figure 7.4. The *interrupt block* code calls the footnote block into force. The superior numeral for the citation is created by changing the point size to five and raising the VideoComp baseline three points: the

Figure 7.3
Format statement for single column text
with provision for elementary footnotes

```
A3 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,470;bx,3,12;
     pd,468;gv89,468;ad,pn,pn,1;fn,1;
     sb,1;lb,0;tb,0;rb,288;bb,468;ju;ib;
     sb,2;lb,0;tb,0;rb,288;bb,468;ps,10;bl,10;tf,6;ce;dc,pn;nl;
     sb,3;ju;cb;
     ab[[pd,480;gv89,456;jv;pg;sb,1;bb,456;ad,pn,pn,1;ib;sb,2;
        ps,10;tf,6;bb,456;Y2;nl;sb,3;ju;cb;by,1,24;by,2,0]];
     wb[[jv;pg;sb,1;bb,456;ad,pn,pn,1;ib;sb,2;ps,10;tf,6;
        bb,456;Y2;nl;sb,3;ju;cb]]]
```

footnote text is to be set in eight point type. Next, the citation is set and finally the baseline is returned to nominal and the point size is changed to eight. The text of the footnote may now be set.

Close footnote format statement

Following the footnote text, the end footnote format statement, shown in Figure 7.4, will position the footnote, shorten the text block bottom boundary, and set the footnote citation in the text block; then the setting of text may be resumed at the point of interruption.

A *new line* code closes the footnote. The vertical setting position after this action is below the last footnote line inclusive of the leading on the last line. The depth in the footnote block is computed by subtracting two points from **cy** (the footnote is set 8/10 or is "leaded" two points) and assigning the result to **gv192.** Activity in the footnote block is now complete and the text block is re-entered.

The proper location for the footnote block is computed by subtracting **gv192** from the number assigned to the *page depth* global variable and is then established by setting the Y-coordinate of block three to the resulting number. See Figure 7.2.

Figure 7.4
Format statements to initiate elementary footnotes and to resume text

Enter footnote block	[ib;
Prepare for citation	ps,5;tf,5;up,96;
Typeset citation	dc,fn;
Prepare for footnote	up,0;ps,8;tf,4]
Set footnote text	...
Note footnote depth	[nl;df,gv192,cy,2;
Resume text block	cb;
Place block three	
compute location	df,gv160,pd,gv192;
fix origin	by,3,gv160;
Compute footnote commitment	ad,gv90,gv192,2;
Adjust text block bottom boundary	df,bb,gv89,gv90;
Store font	gv2,ps;gv4,tf;
Prepare for citation	ps,5;tf,5;up,128;
Typeset citation	dc,fn;
Prepare for text	up,0;ps,gv2;tf,gv4;
Increment citation counter	ad,fn,fn,1]

The depth which has been commited to footnotes must be computed in order to shorten the text block bottom boundary to account for the space occupied by the footnote. This space is the sum of the footnote depth (excluding last line lead—already stored in **gv192**) and the white space specified between text and footnote by the style specification.

One might ask: why subtract two points and then add them back? In the interest of generality; the footnote leading and the white space between text and footnotes might not always be equal. This distinction will be more readily apparent when the format statements are generalized in Chapter Ten. The footnote space allocation is stored in **gv90**. The text block bottom boundary is shortened by subtracting the footnote allocation from the standard depth of the nominal text block.

Preparation for setting the footnote citation in the text block begins with the storage of the point size and typeface in force. This is done because the footnote citation may have come in secondary text as well as in primary text and the typeface will be changed to the Size Range I font for the footnote citation. The superior numeral is again simulated by changing the point size to five, changing the typeface to number five, and raising the baseline. In this case the baseline is raised five points. This amount is assumed acceptable for either 10 point primary or 9 point secondary text in order to simplify this example. One can be more precise by subtracting five points from the text block point size in force at the point of the footnote citation, multiplying the resulting difference by 32, and finally by raising baseline by that amount.

The citation is set, the point size, typeface, and baseline are restored, and the footnote counter is incremented. The setting of primary text may now continue.

FOOTNOTE CARRYOVER

THE PROVISIONS NECESSARY to accommodate footnote carryover are: detect that carryover has occurred, set the remainder of the footnote in the suspense block, and finally, if additional footnotes appear on the page which has a carryover, reposition the suspense block.

Carryover detection

The sample footnote shown in Figure 7.5 could be handled by the format statements already developed because the footnote fits. In order to detect the carryover condition, block three will be given a bottom boundary. The start footnote procedure will compute the amount of space already committed to text at the citation and add to this the depth specified to separate text and footnotes.

Following these computations, the footnote block will be entered and the remainder of the space in the nominal text block allotment will be assigned to block three. Finally the start footnote procedure will set the citation in the footnote block as before. The format statement is shown in Figure 7.6.

Example: footnote fits

In the example of Figure 7.5, if the citation is encountered in the twenty-eighth line and only primary text has been set to the citation point, the depth committed is:

$$(27 \times 12) + 10 + 2 = 336 \text{ points;}$$

that is, 28 lines minus two points for leading in the 28th line plus 2 points for the space to be provided between text and footnotes. The bottom boundary of the footnote block will be set to:

df,bb,gv89,gv90
gv89 − gv90 = bb

$$456 - 336 = 120 \text{ points,}$$

If the footnote in Figure 7.5 requires only two lines, the value of **cy** following the new line code which opens the end footnote format statement will be 20. The depth of the footnote is:

df,gv192,cy,2
cy − 2 = gv192

$$20 - 2 = 18 \text{ points,}$$

and the origin of block three is:

df,160,pd,gv192
pd − gv192 = gv160

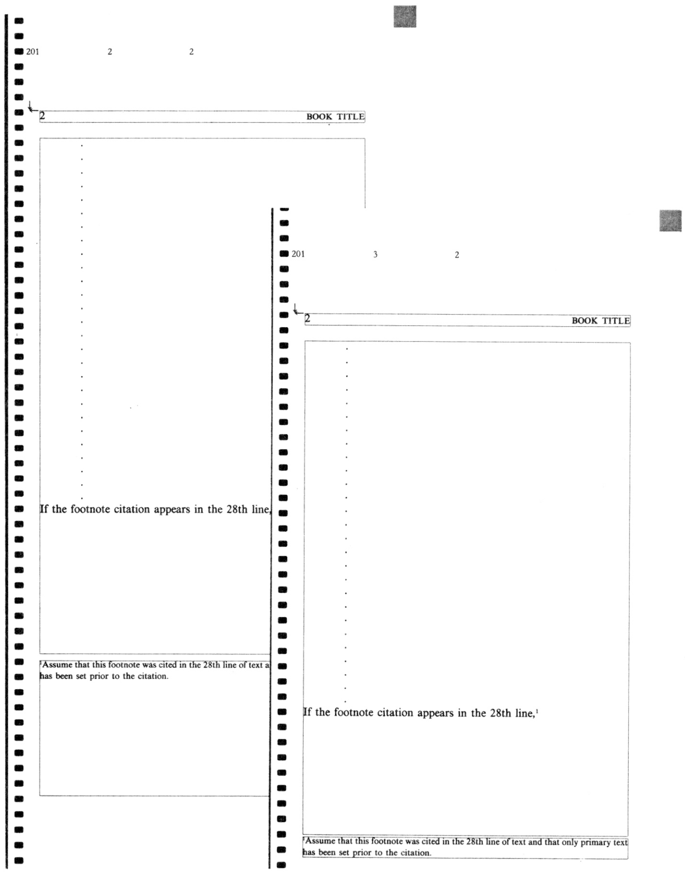
$$480 - 18 = 462 \text{ points.}$$

The non-text commitment is the footnote depth plus the text/footnote white space:

ad,gv90,gv192,2
gv192 + 2 = gv90

$$18 + 2 = 20 \text{ points,}$$

Figure 7.5
Sample footnote



If the footnote citation appears in the 28th line,[1]

[1]Assume that this footnote was cited in the 28th line of text and that only primary text has been set prior to the citation.

Figure 7.6
Format statement to initiate footnote and to resume text

Compute text commitment	[ad,gv90,cy,ps; ad,gv90,gv90,2;
Enter footnote block	ib;
Fix bottom boundary	df,bb,gv89,gv90;
Prepare for citation	ps,5;tf,5;up,96;
Typeset citation	dc,fn;
Prepare for footnote	up,0;ps,8;tf,4]
Set footnote text	...
Terminate line	[nl;
Record depth	
block three	eq,gv182,0[[df,gv192,cy,2]];
block zero	eq,gv182,1[[df,gv193,cy,2]];
Resume text block	cb;
Place block three	
compute location	df,gv160,pd,gv192;
fix origin	by,3,gv160;
Compute footnote commitment	ad,gv90,gv192,2;
Adjust bottom boundary	df,bb,gv89,gv90;
Store font	gv2,ps;gv4,tf;
Prepare for citation	ps,5;tf,5;up,128;
Typeset citation	dc,fn;
Prepare for text	up,0;ps,gv2;tf,gv4;
Increment citation counter	ad,fn,fn,1]

and the bottom boundary of the text block is set to:

df,bb,gv89,gv90
gv89 − gv90 = bb
456 − 20 = 436 points.

Example: footnote carryover

In the example of Figure 7.5, the footnote fit. If the citation for the footnote is encountered in the thirty-seventh line and only primary text has been set in the text block prior to the point of citation—as shown in Figure 7.9—the text block and the text/footnote white space commitment is:

$$(36 \times 12) + 10 + 2 = 444 \text{ points,}$$

and the bottom boundary of the footnote block will be:

df,bb,gv89,gv90
gv89 − gv90 = bb
456 − 444 = 12 points.

After the first line of the footnote is set, the bottom boundary of block three will be encountered, and the first requirement for handling carryover—detection—will have been accomplished. The block three bottom boundary procedure, shown in Figure 7.7, first records the depth of block three. In this case it is eight points; **cy**, when the bottom boundary has been encountered omits the last line lead.

After carryover detection

After recording block three depth, the procedure starts the suspense block. This action assigns all of the block three attributes to block zero; the bottom boundary then is 12 points. The bottom boundary is changed to the maximun text block depth (no text has yet been set on the next page) and finally sets the carryover indicator to one.

The remaining line of the footnote is now set in block zero. The processor will save this block for the next page. Its position on the next page will be established in the next execution of the text block bottom boundary procedure. But first, following the footnote, the end footnote format statement appears in input; the modification to it will now be considered.

New close footnote procedure

The footnote just concluded carried over; the initial fragment of the footnote is in block three, the depth of block three is recorded in **gv192**, and the carryover indicator, **gv182**, has the value one. The end footnote format statement of Figure 7.4 is modified as follows: on the condition that **gv182** has the value zero, depth (of block three) is recorded in **gv192** and on the condition that **gv182** has the value one, depth (of block zero) is recorded in **gv193**.

Figure 7.7
Footnote block bottom boundary procedure

```
Depth of fragment          [gv192,cy;
Start carryover block       sb,0;
Fix bottom boundary         bb,gv89;
Set carryover indicator     gv182,1]
```

The remainder of the statement, shown in its latest form in Figure 7.6, is the same as before. The depth of block three, recorded in **gv192** during execution of the block three bottom boundary procedure, is eight. The depth of the carryover, recorded in **gv193** is also eight. The origin of block three is:

df,gv160,pd,gv192
pd − gv192 = gv160
480 − 8 = 472 points.

The non-text commitment is the footnote depth plus the text/footnote white space:

ad,gv90,gv192,2
gv192 + 2 = gv90
8 + 2 = 10 points,

and the bottom boundary of the text block is set to:

df,bb,gv89,gv90
gv89 − gv90 = bb
456 − 10 = 446 points.

Note that when a footnote carries over, the line of citation in text becomes the last line of text on the page.

Figure 7.8
Additions to text block bottom boundary procedure
to position footnote carryover fragment

```
                                        .
                                        .
                                        .
Reset depth                   gv194,0;
Test carryover                eq,gv182,1[[
Place carryover block
          compute origin                    df,gv161,pd,gv193;
              fix origin                    by,0,gv161;
Record carryover depth                      ad,gv194,gv193,2;
Compute footnote
             commitment                     ad,gv90,gv193,2;
Adjust bottom boundary                      df,bb,gv89,gv90;
Reset carryover indicator                   gv182,0]];
                                        .
                                        .
                                        .
```

Modification to text block bottom boundary procedure

The text block bottom boundary procedure will now be augmented to position the suspense block on the next page. The additions are shown in Figure 7.8. They take the form of a procedure conditional on **gv182** having the value one.

First, however, consider that: footnotes which appear on the page which has a carryover must follow the carryover fragment; block zero must, in this case, be moved up. Block zero depth will be recorded in **gv194** as an input to the repositioning and as evidence that a page being set has a carryover. For this later purpose, **gv194** is first initialized to zero.

Then, if **gv182** is equal to one, there is a carryover which must be positioned; depth of the carryover was assigned to **gv193** in the footnote block bottom boundary procedure. The origin of block zero is:

df,gv161,pd,gv193

$$\mathbf{pd} - \mathbf{gv193} = \mathbf{gv161}$$

$$480 - 8 = 472 \text{ points.}$$

for purposes of repositioning block zero if further footnotes appear on this page, the last line leading must be added back in when the carryover depth is assigned to **gv194**:

ad,gv194,gv193,2

$$\mathbf{gv193} + \mathbf{2} = \mathbf{gv194}$$

$$8 + 2 = 10 \text{ points.}$$

The non-text committment is the suspense block depth plus the text/footnote white space:

ad,gv90,gv193,2

$$\mathbf{gv193} + \mathbf{2} = \mathbf{gv90}$$

$$8 + 2 = 10 \text{ points,}$$

and the bottom boundary of the text block is set to:

df,bb,gv89,gv90

$$\mathbf{gv89} - \mathbf{gv90} = \mathbf{bb}$$

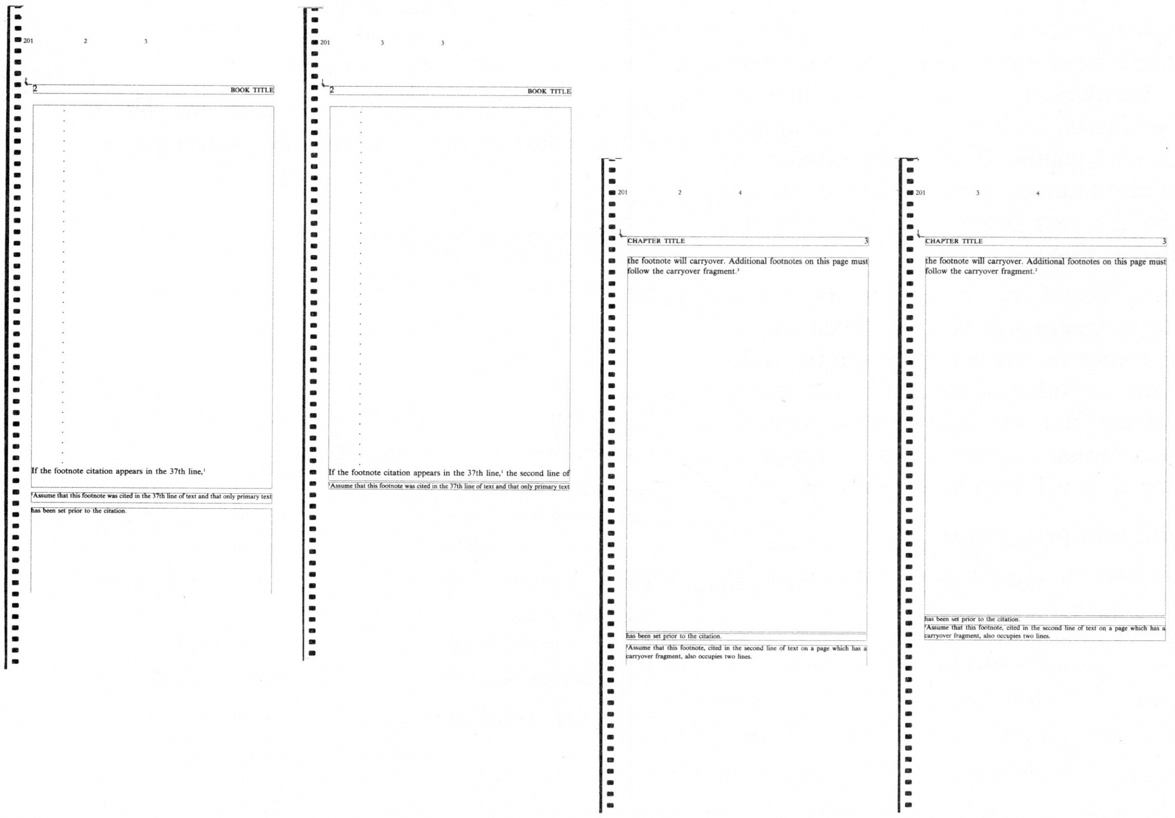
$$456 - 10 = 446 \text{ points.}$$

Finally the carryover indicator is reset to zero and the conditional procedure is closed. The modification may be noted in the bottom boundary procedures in Figure 7.10; they are set in boldface type.

Additions to footnote procedures

Finally, additions to the start footnote and end foot-

Figure 7.9
Example of footnote carryover



If the footnote citation appears in the 37th line,[1]

[1] Assume that this footnote was cited in the 37th line of text and that only primary text

has been set prior to the citation.

If the footnote citation appears in the 37th line,[1] the second line of

[1] Assume that this footnote was cited in the 37th line of text and that only primary text

the footnote will carryover. Additional footnotes on this page must follow the carryover fragment.[2]

has been set prior to the citation.

[2] Assume that this footnote, cited in the second line of text on a page which has a carryover fragment, also occupies two lines.

the footnote will carryover. Additional footnotes on this page must follow the carryover fragment.[2]

has been set prior to the citation.

[2] Assume that this footnote, cited in the second line of text on a page which has a carryover fragment, also occupies two lines.

note procedures will provide for moving block zero if additional footnotes are encountered on the page which has a carryover fragment from the preceding page. The depth of the carryover fragment must be considered when the "space-already-committed" is computed in the start footnote procedure. Depth of the fragment was assigned to **gv194** in the bottom boundary procedure.

When the second footnote is encountered in the second line of text in the carryover example of Figure 7.9, the depth committed is:

(1 × 12) + 10 + 2 = 24 points,
ad,gv90,gv90,gv194
gv90 + **gv194** = **gv90**
48 + 10 = 58 points,

and the bottom boundary of block three will be set to:

df,bb,gv89,gv90
gv89 − **gv90** = **bb**
456 − 34 = 422 points.

The addition to the start footnote procedure is shown in boldface type in Figure 7.10.

After the two-line footnote has been set in block three, the value of **cy** is 20. The depth of the footnote is:

df,gv192,cy,2
cy − **2** = **gv192**
20 − 2 = 18 points,

and the origin of block three is:

df,gv160,pd,gv192
pd − **gv192** = **gv160**
480 − 18 = 462 points.

The block three commitment to footnotes is the footnote depth plus the text/footnote white space:

ad,gv90,gv192,2
gv192 + **2** = **gv90**
18 + 2 = 20 points.

At this point the depth of the carryover fragment is added to the non-text committment:

ad,gv90,gv90,gv194
gv90 + **gv194** = **gv90**
20 + 10 = 30.

The origin of the suspense block is recomputed and fixed:

df,gv161,gv160,gv194
gv160 − **gv194** = gv161
462 − 10 = 452 points,

and the conditional procedure is ended.

Now the bottom boundary of the text block—as before—is set to:

df,bb,gv89,gv90
gv89 − **gv90** = **bb**
456 − 30 = 426.

The conditional procedure is shown in boldface type in the end footnote format statement in Figure 7.10.

Exercise: The characters: *, †, ‡, §, and ¶ are sometimes used as footnote citations when the use of Arabic superscript numerals is likely to cause confusion. They are normally used full size and on the same baseline as text; each page starts the sequence anew. Modify the format statements of Figure 7.10 to use these characters as footnote citations. (Hint: use a series of conditional procedures based on the value of the footnote citation counter.)

FOOTNOTES FOR TWO COLUMN TEXT

ANOTHER METHOD of handling footnote carryover, when the footnotes in the text to be typeset are always short, is to allow that portion of the footnote which will not fit, to hang below the nominal text depth. This treatment allows the typesetter to avoid one footnote

Figure 7.10
Format statements for single column text and footnotes

```
C1 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,470;bx,0,12;bx,3,12;
      pd,468;gv89,468;gv194;gv182,0;ad,pn,pn,1;fn,1;

      sb,1;lb,0;tb,0;rb,288;bb,468;ps,10;bl,12;tf,4;ju;ib;

      sb,2;lb,0;tb,0;rb,288;bb,468;ps,10;bl,10;tf,6;ce;dc,pn;nl;

      sb,3;ju;wb[[gv192,cy;sb,0;bb,468;gv182,1]];cb;

      ab[[pd,480;jv;pg;sb,1;bb,456;gv194,0;
          eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                      ad,gv194,gv193,2;ad,gv90,gv193,2;
                      df,bb,gv89,gv90;gv182,0]];
          ad,pn,pn,1;ib;sb,2;ps,10;tf,6;
          bb,456;Y2;nl;sb,3;ju;cb;by,1,24;by,2,0]];
      wb[[jv;pg;sb,1;bb,456;gv194,0;
          eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                      ad,gv194,gv193,2;ad,gv90,gv193,2;
                      df,bb,gv89,gv90;gv182,0]];
          ad,pn,pn,1;ib;sb,2;ps,10;tf,6;
          bb,456;Y2;nl;sb,3;ju;cb]]]


F4 ≡ [ad,gv90,cy,ps;ad,gv90,gv90,2;ad,gv90,gv90,gv194;
      ib;df,bb,gv89,gv90;ps,5;tf,5;up,96;dc,fn;up,0;ps,8;tf,4]

F5 ≡ [nl;eq,gv182,0[[df,gv192,cy,2]];
        eq,gv182,1[[df,gv193,cy,2]];
      cb;df,gv160,pd,gv192;by,3,gv160;ad,gv90,gv192,2;
      ne,gv194,0[[ad,gv90,gv90,gv194;
                  df,gv161,gv160,gv194;by,0,gv161]];
      df,bb,gv89,gv90;gv2,ps;gv4,tf;ps,5;tf,5;up,128;
      dc,fn;up,0;ps,gv2;tf,gv4;ad,fn,fn,1]
```

carryover case that is considered bad typography: the case in which a footnote carries overleaf from a right-hand page to the following left-hand page.

A style specification for two-column text with two-column footnotes—incorporating the hanging carryover footnote—is given in Figure 7.11. Footnote format statements to satisfy this specification will be developed and the two-column text algorithm of Chapter Five will be modified to accommodate them.

As with the single-column footnote case, two format statements—one to precede the footnote and one to follow it will be developed. The footnote text will be entered into input at the point of citation in text. The format statements will be developed in two steps:

- First—assuming that the footnote will fit in the column in which it is cited, statements to initiate footnote setting and to resume text will be developed and
- Second—the carryover case will be considered and the format statements will be modified.

The elementary—footnote fits—case

First the format statement for two-column text—A4—must be modified in anticipation of the appearance of footnotes. The modified format statement is shown in Figure 7.13 with the additions set out in boldface type. Footnotes will be set in blocks three and five as shown in the block map of Figure 7.12; the X-coordinates of these two blocks are set to 12 and 204 respectively to be consistent with the corresponding primary text blocks. The *page depth* global variable is assigned the value 550 points for use in computing footnote block position.

General variable three will be manipulated in such a way that its assigned value will be three or five to correspond with the identity of the column-one and column-two footnote blocks, respectively. Its value is initialized to three—the column one footnote block number. (Aside from the footnote problem under consideration, the dedicated synonym name Y2 is embedded in the pagination procedure to provide for the folio treatment.) Following typesetting of the folio, block three is started and its boundaries are stated.

Figure 7.11
Style specification

.
.
.

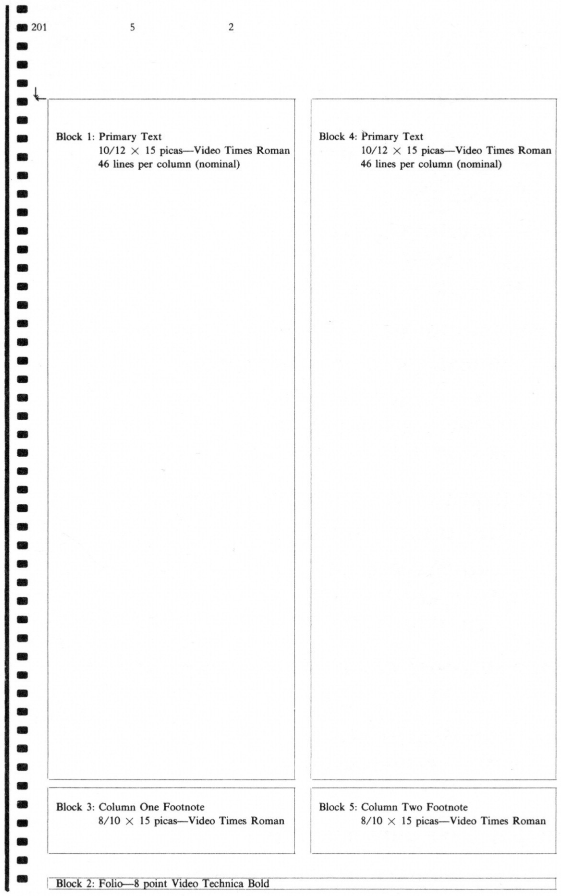
Footnotes	two-column 8/10 × 15 picas, Video Times Roman, cited by consecutive Arabic numerals, first footnote in each chapter is numbered 1, minimum 4 points above, footnotes carry from LH to RH column on a page, carryover in RH column is allowed to hang below nominal text depth

.
.
.

In the column-one *when bottom* procedure, the column-two footnote block is started and **gv3** is assigned the value five. Should a footnote be encountered during the typesetting of text in column-two, block five is ready.

In the column-two *when bottom* procedure, the folio block is started and its attributes are stated since the interrupting block prior to this action is block five—the column-two footnote block. After the folio and running foot are typeset, the column-one footnote block is started and its attributes are stated. Finally, **gv3** is assigned the value three. Should a footnote be

Figure 7.12
Two column text with footnotes


201
5
2
Block 1: Primary Text
10/12 × 15 picas—Video Times Roman
46 lines per column (nominal)
Block 4: Primary Text
10/12 × 15 picas—Video Times Roman
46 lines per column (nominal)
Block 3: Column One Footnote
8/10 × 15 picas—Video Times Roman
Block 5: Column Two Footnote
8/10 × 15 picas—Video Times Roman
Block 2: Folio—8 point Video Technica Bold

encountered during the typesetting of column-one, block three is ready.

The format statements for the elementary two-column footnotes are shown in Figure 7.13. The start footnote format statement enters the footnote block and sets the footnote citation. The footnote block, as provided by the most recent execution of the *when bottom* boundary procedure, will be three or five depending upon the text column in which the citation is encountered. Note that the format statement is precisely the same as the one shown in Figure 7.4 for starting elementary single-column footnotes.

Following the footnote text, the end footnote format statement will position the footnote, shorten the appropriate text column, and typeset the citation in the text block. The setting of text may then be resumed. The only essential difference between the end footnote format statements of Figures 7.13 and 7.4 is in the statement which establishes the Y-coordinate of the footnote block origin. In Figure 7.4, the footnote block is always block three. In Figure 7.13, it is either block three or block five; the identity of the current footnote block is maintained in **gv3**.

Detection of footnote carryover

In order to detect the footnote carryover condition if it occurs, the start footnote procedure will be modified to compute the amount of space comitted to text when the footnote citation is encountered and then to add to this comittment the white space specified to separate text and footnotes. Following this computation, the footnote block will be entered and all of the space remaining in the nominal text column allocation will be assigned to the footnote block. These additions are shown in the final start footnote procedure given in Figure 7.14; they are set out in boldface type.

Figure 7.13
Format statement for two-column text
with elementary footnotes

```
A7 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,568;bx,3,12;
      bx,4,204;by,4,0;bx,5,204;gv1,0;gv3,3;
      pd,550;ad,pn,pn,1;fn,1;

      sb,1;lb,0;tb,0;rb,180;bb,550;ps,10;bl,12;tf,4;ju;ib;

      sb,2;lb,0;tb,0;rb,372;ps,8;bl,8;tf,8;Y2;nl;

      sb,3;lb,0;rb,180;cb;
      wb[[eq,gv1,0[[jv;sb,4;ib;sb,5;gv3,5;cb;gv1,2]];
          eq,gv1,1[[jv;pg;sb,1;ad,pn,pn,1;ib;
                   sb,2;lb,0;rb,372;ps,8;tf,8;Y2;nl;
                   sb,3;lb,0;rb,180;ps,8;gv3,3;cb]];
          df,gv1,gv1,1]]]

F8 ≡ [ib;ps,5;tf,5;up,96;dc,fn;up,0;ps,8;tf,4]

F9 ≡ [nl;df,gv192,cy,2;cb;df,gv160,pd,gv192;by,3,gv160;
      ad,gv90,gv192,4;df,bb,550,gv90;gv2,ps;gv4,tf;
      ps,5;tf,5;up,128;dc,fn;up,0;ps,gv2;tf,gv4;ad,fn,fn,1]
```

A bottom boundary procedure for the footnote block

As in the earlier, single-column footnote format statement, the footnote block *when bottom* procedure will record the depth of the footnote at the point of bottom boundary encounter. This information is required whether the text column is one or two; it will be used subsequently in the end footnote procedure to position the footnote block—three or five again depending upon which column is involved.

Next, the bottom boundary of the footnote block is extended to 550 points. If the footnote started in column one (ie., in block three), the remainder of the footnote will be set in block five and no text has yet been set in column two (ie., in block four)—the entire text column depth is available to conclude the footnote. If the footnote started in column two (ie., in block five), the footnote will be concluded in block five —the hanging footnote carryover. In either case, the primary consideration is to get the footnote block bottom boundary far enough removed that it will not be encountered before the end of the footnote. And, the original assumption upon which these rules are based is that the footnotes to be typeset are short. Finally, if the text column is one, (**gv1** is zero if this is true) block five is started. The block three *when bottom* procedure appears in line seven of Figure 7.14; it is set in boldface type.

Modifications to the close footnote format statement

At the end of a footnote, two conditions are possible: the footnote fit or it did not. If the footnote fit in the available space, the elementary end footnote format statement of Figure 7.13 is sufficient to end the footnote. In order to handle the "footnote fit" case while still providing for the possibility that the footnote might not always fit, the recording of footnote depth (in **gv192**) is made conditional on **gv182** having the value zero; see the first line of **F9** in Figure 7.14.

If the footnote did not fit, the footnote bottom boundary was encountered and the depth of the "fragment-that-fit" the available space has already been assigned to **gv192**. In this event, **gv182** has the value one and the depth of the current footnote block is assigned to **gv193**. This is accomplished in line two of

F9 in Figure 7.14 by the conditional statement. Note here that: if the footnote started in column one, **gv192** has a value equal to the depth of block three and **gv193** has a value equal to the depth of block five. If the footnote started in column two, **gv192** and **gv193** are both measures of block five: **gv192** has a value equal to the depth of the portion of the footnote which fits the nominal text block depth (and will be used to position block five) and **gv193** has a value equal to the total depth of block five. In the later case, **gv193** will not be used. These changes are set in boldface type in Figure 7.14.

Positioning the footnote blocks

If the footnote fit in the available space—either in column one or column two, the final end footnote procedure of Figure 7.14 reduces to the elementary statement of Figure 7.13—**gv182** is equal to zero. The depth of the footnote block is assigned to **gv192**, the Y-coordinate of the footnote block origin is computed as the difference between **pd** and **gv192** and is assigned to **gv160**. The footnote block origin is then fixed via the statement **..;by,gv3,gv160;..**. The identity of the footnote block is maintained in **gv3**. Next the footnote commitment is computed and assigned to **gv90** and the text block bottom boundary is adjusted by the statement **..;df,bb,550,gv90;.** The statement works in either case: if the footnote block is three, the text block is one and its bottom boundary is adjusted. If the footnote block is five, the text block is four and its bottom boundary is adjusted.

Carryover footnote in column one

Assume that the footnote was cited in text column one; the footnote was set in block three and **gv3** has the value three. When the footnote block bottom boundary is encountered, the depth of block three is assigned to **gv192** and the footnote is concluded in block five. At the footnote conclusion, the depth of block five is assigned to **gv193**—the carryover indicater, **gv182**, has the value one. Further, since the text column is one, **gv1** has the value zero. The Y-coordinate of block five is computed as the difference between **pd** and **gv193**. Computation of the column two footnote commitment follows and the value is assigned to **gv91** (for subsequent use in the text block *when bottom* procedure). Then the code **cb** resumes the text block which is still block one and the Y-coordinate of block three is computed as the difference between **pd** and **gv192**. The origin of block three is then assigned by the statement **..;by,gv3,gv160;..** and the bottom boundary of block one is adjusted.

Carryover in column two

Assume now, that the footnote was cited in text column two; the footnote is set in block five and **gv3** has the value five. When the footnote block bottom boundary is encountered, the depth—at that point—in block five is assigned to **gv192** and setting continues in block five. At the conclusion of the footnote, the total depth of block five is assigned to **gv193**. (This measurement is not needed and will not be used; but recording it requires less coding than not recording it would.) Since the text column is two, **gv1** has the value one and the inner conditional statement is not executed. The code **cb** resumes the text block, which is four, and the Y-coordinate of block five is computed as the dfference between **pd** and **gv192**. The origin of block five is then assigned by the statement **..;by,gv3,gv160;..** and the bottom boundary of block four is adjusted.

Modifications to the pagination algorithm

Finally the ring is closed by three modifications to the column-one and one modification to the column-two portions of the text block *when bottom* procedure. First—the bottom boundary of block four is set to 550 points since it may have been shortened on the preced-

Figure 7.14
Format statements for two column text with footnotes

```
A7 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,268;bx,3,12;
      bx,4,204;by,4,0;bx,5,204;
      gv1,0;gv3,3;pd,550;ad,pn,pn,1;fn,1;

      sb,1;lb,0;tb,0;rb,180;bb,550;ps,10;bl,12;tf,4;ju;ib;

      sb,2;lb,0;rb,372;ps,8;bl,10;tf,8;Y2;nl;

      sb,3;lb,0;rb,180;
      wb[[gv192,cy;bb,550;gv182,1;eq,gv1,0[[sb,5]]]];cb;

      wb[[eq,gv1,0[[jv;sb,4;bb,550;eq,gv182,0[[ib;sb,5;cb]];
                       eq,gv182,1[[df,bb,550,gv91;gv182,0]];
                       gv3,5;gv1,2]];
          eq,gv1,1[[jv;pg;sb,1;bb,550;ad,pn,pn,1;ib;sb,2;
                       lb,0;rb,372;ps,8;tf,8;Y2;nl;sb,3;lb,0;
                       rb,180;ps,8;cb;gv3,3]];
          df,gv1,gv1,1]]]

F8 ≡ [ad,gv90,cy,ps;ad,gv90,gv90,4;
      ib;df,bb,550,gv90;ps,5;tf,5;up,96;dc,fn;up,0;ps,8;tf,4]

F9 ≡ [nl;eq,gv182,0[[df,gv192,cy,2]];

         eq,gv182,1[[df,gv193,cy,2;
                       eq,gv1,0[[df,gv161,pd,gv193;by,5,gv161;
                                   ad,gv91,gv193,4]]]];
      cb;df,gv160,pd,gv192;by,gv3,gv160;
      ad,gv90,gv192,4;df,bb,550,gv90;gv2,ps;gv4,tf;
      ps,5;tf,5;up,128;dc,fn;up,0;ps,gv2;tf,gv4;ad,fn,fn,1]
```

ing page if that page had footnotes in column two. If there was no footnote carryover, the footnote block is still number three and block five must be started. The codes which start block five appear in a conditional procedure. Then: if there has been a footnote carryover, block five was started in the execution of the footnote block *when bottom* procedure and the amount of column-two depth comitted to footnotes was computed and assigned to **gv91** in the end footnote format statement. The value of **gv182** is one; it is necessary only to adjust the value of the text block (block four) bottom boundary and reset the carryover indicator to zero.

The column two portion of the text block *when bottom* procedure must reset the block one bottom boundary to 550 points since it may have been changed on the preceding page if that page had footnotes in column one.

Exercise: Modify the footnote block *when bottom* procedure in the pagination algorithm of Figure 7.14 to provide a short rule (32 points long and starting at the left edge of the column) above the concluding fragment of a footnote carried over from column one to column two.

Answers:—

```
F4 ≡ [ad,gv90,cy,ps;ad,gv90,gv90,2;ad,gv90,gv90,gv194;
      ib;df,bb,gv89,gv90;eq,fn,1[*];eq,fn,2[†];
      eq,fn,3[‡];eq,fn,4[§];eq,fn,5[¶]]
F5 ≡ [nl;eq,gv182,0[[df,gv192,cy,2]];
         eq,gv182,1[[df,gv193,cy,2]];
      cb;df,gv160,pd,gv192;by,3,gv160;ad,gv90,gv192,2;
      ne,gv194,0[[ad,gv90,gv90,gv194;
                  df,gv161,gv160,gv194;by,0,gv161]];
      df,bb,gv89,gv90;eq,fn,1[*];eq,fn,2[†];
      eq,fn,3‡];eq,fn,4[§];eq,fn,5[¶[fn,0]];ad,fn,fn,1]
...
         wb[[gv192,cy;bb,550;gv182,1;eq,gv1,0[[sb,5;rb,32;fl,—;dn,8]]]]
```

8

PROCEDURES TO ALLOCATE AND TO POSITION WHITE SPACE

MANY BOOKS—technical books in particular—contain illustrations, charts, tables, or diagrams. One method of handling this requirement with Page–1 is to leave appropriate white space in the typeset pages so that the artwork can be added before final reproduction proofs are made. This approach minimizes manual effort in the preparation of the "camera ready" reproduction proofs; proper positioning of the running heads and footnotes and other typographic features are provided by the Page–1 processor. This approach requires that the artwork must be complete with titles, captions, and legends; the processor will simply allocate and position a specified depth of white space.

The procedure will position the white space by adjusting text block origin and bottom boundary according to the rules stated for the positioning of artwork; the depth of the illustration for which white space is to be provided will be assigned to a general variable immediately preceding the format statement call in input. Rules for the placement of white space are:

- If the white space to be provided will fit on the page being composed when the citation is encountered, the allocation shall be made on the current page;
- If, at the citation, there is insufficient depth remaining on the page being composed to accommodate the requirement, the allocation shall be made on the following page;
- If the white space to be provided is less than one-half of the current text column depth, the white space shall be provided at the top of the text column;
- If the white space to be provided is greater than one-half of the current text column depth, the white space shall be provided at the bottom of the text column (ie., above footnotes which may be on the page); and
- If the allocation is to be made on the following page, the placement is subject to the same "top-of-column" or "bottom-of-column" rules.

A procedure will be developed to make the decisions stated in the rules for placement and to compute the new block origin and bottom boundary necessary to provide the specified white space. If the white space fits on the citation page, the new block origin and bottom boundary will be stated within the procedure. If the white space is to be carried forward to the next page, the value of the block origin and bottom boundary will be stored in general variables for use when next bottom boundary procedure is executed.

The algorithm for single column text with footnotes developed in Chapter Seven (Figure 7.10) is reproduced in Figure 8.1. Note that the statements of the bottom boundary (in lines 6 and 12 in the bottom boundary procedures) are made with **gv86** rather than an explicitly stated 456 points as in Figure 7.10. The boundary statements are set in boldface type in Figure 8.1. The value of **gv86** will be computed in the "white space" procedure (as will the value of the text block origin) when white space is to be carried forward to the succeeding page. The pagination algorithm of Figure 8.1 will be further modified to include the statement of block origin and other housekeeping matters after the white space procedure has been developed.

The white space procedure

The procedure for placing white space will comprise a number of conditional procedures to comply with the stated style rules. The first level conditional procedures ask the question: Is white space already provided on the next page? General variable 195 will be used as the switch; if **gv195** is equal to zero, white space has not yet been assigned to the next page. If space is assigned to the next page, **gv195** will be set to one.

Art not yet assigned to the next page

The procedure for allocating and positioning white space is shown in Figure 8.2 in flowchart form and in Figure 8.3. Assume that a "graphic citation" is to specify ten lines of white space and that text is being set 10/12. The citation for the desired white space is **[gv6,120;c9]** in which **gv6** is set equal to 120 points (ten-10/12 lines) and **c9** is the format label given the white space procedure. Assume **gv195** is equal to zero. The first step is to compute the space remaning in the text block. This is done by computing the space already comitted in the text block—through the line in which the citation was encountered—and then subtracting this value from that of the bottom boundary. If **cy** has the value zero, the white space citation has been encountered in the first line in the block. If **cx** also has the value zero, the citation is the first bit of input encountered in the text block—set **gv90** to zero. No text block space has yet been comitted. If **cx** has a value other than zero, the citation has been encountered in the first line in the text block and the depth comitted is equal to the point size in force. See line two of Figure 8.3.

If **cy** has a value greater than zero and if **cx** is equal to zero, the depth comitted is less than **cy** by the amount of interline lead. The value of **gv90** is computed as **cy** minus **bl** plus **ps**. If **cx** is not equal to zero, the citation has been encountered within a line of text and **gv90** is computed as **cy** plus **ps**. This calculation is in line three of Figure 8.3.

Finally, the value of **gv90** is subtracted from the current value of the bottom boundary and the resulting difference is assigned to **gv90**; that is, **gv90** now is set equal to the uncomitted space in the current text block. See line three of Figure 8.3. (Note that if the white space citation is encountered in a line which was initiated by a **dn** or a **dj** command, the value of **gv90** is less than space remaining by the point size. This circumstance will likely not occur; indeed it can be avoided in markup.)

With the space remaining on the current page known, the next step is to determine whether the white space requested will fit the current page or whether it must be placed on the following page. If the depth remaining (**gv90**) is greater than or equal to the space requested (assigned to **gv6**), the space will be allocated on the current page. If **gv90** is less than **gv6**, the space must be allocated on the next page, these two conditional procedures are on lines four and five of Figure 8.3.

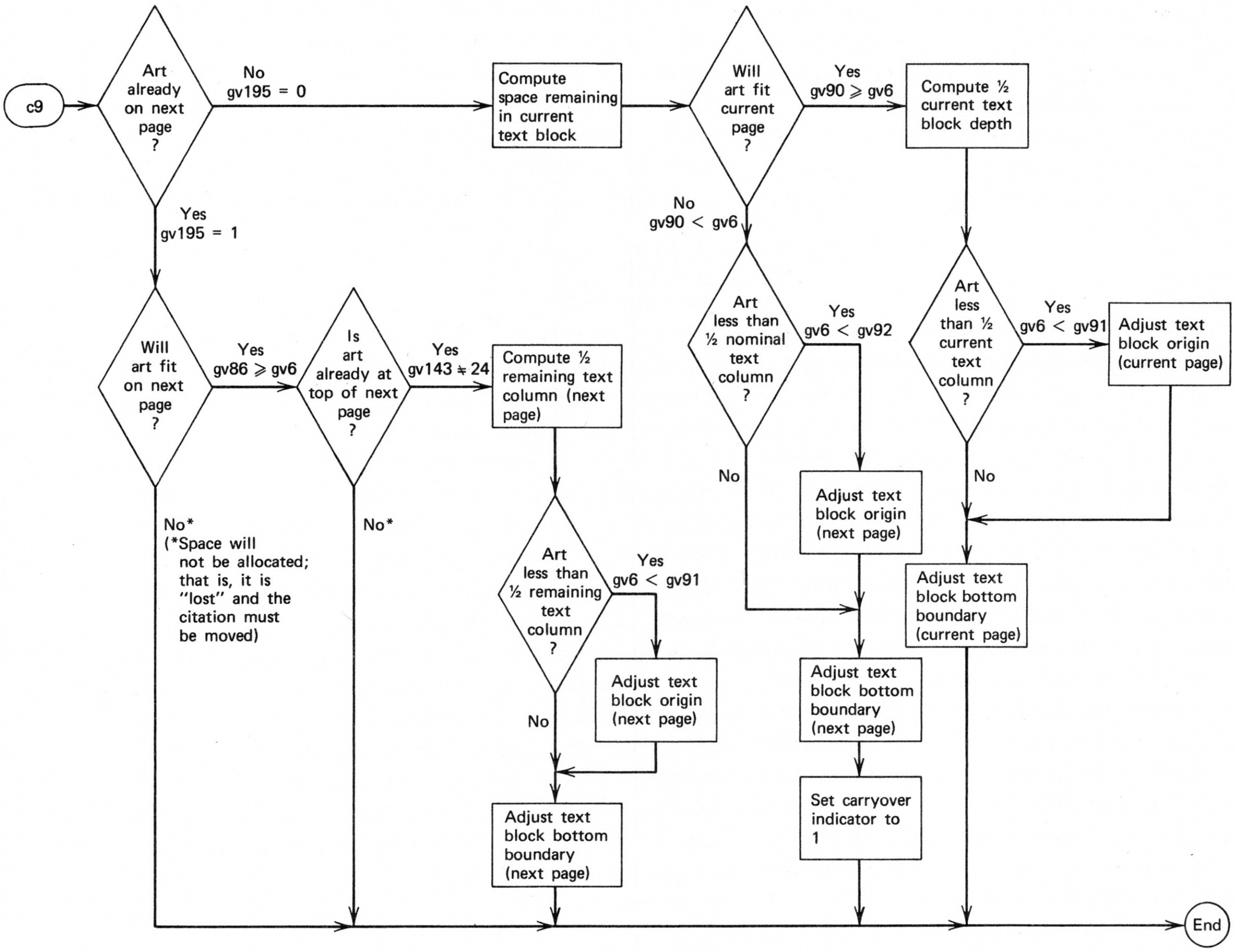
Sufficient space on the current page

If **gv90** is greater than or equal to **gv6** the space will be allocated on the current page subject to the relationship between the depth of white space requested and depth of the current text column. The current value of the bottom boundary is divided by two and the result is assigned to **gv91**. If **gv6** is less than **gv91** (that is, less than one-half the requested text column), the white space will be positioned at the top of the text column by moving the text block origin downward and by shortening the text block bottom boundary, both by the value assigned to **gv6**. If **gv6** is greater than or equal to **gv91,** the white space will be positioned at the bottom of the text column by shortening the text block bottom boundary. Only one conditional procedure is necessary since the bottom boundary is shortened in either case and only the bottom boundary is altered in the latter case. The change in text block origin is made conditional on the value of **gv6** being less than that of **gv91**. See line four of Figure 8.3.

Figure 8.1
Format statement for single column text with footnotes

```
c5 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,470;bx,0,12;bx,3,12;pd,468;
      gv89,468;gv86,468;gv194,0;gv182,0;ad,pn,pn,1;fn,1;

      sb,1;lb,0;tb,0;rb,288;bb,468;ps,10;bl,12;tf,4;ju;ib;

      sb,2;lb,0;tb,0;rb,288;bb,468;ps,10;bl,10;tf,6;y4;nl;

      sb,3;ju;wb[[gv192,cy;sb,0;bb,468;gv182,1]];cb;

      ab[[gv89,456;pd,480;jv;pg;sb,1;bb,gv86;gv194,0;
          eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                      ad,gv194,gv193,2;ad,gv90,gv90,2;
                      df,bb,gv89,gv90;gv182,0]];
          ad,pn,pn,1;ib;sb,2;ps,10;tf,6;bb,gv86;Y2;nl;
          sb,3;ju;cb;by,1,24;by,2,0]];
      wb[[jv;pg;sb,1;bb,gv86;gv194,0;
          eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                      ad,gv194,gv193,2;ad,gv90,gv90,2;
                      df,bb,gv89,gv90;gv182,0]];
          ad,pn,pn,1;ib;sb,2;ps,10;tf,6;
          bb,gv86;Y2;nl;sb,3;ju;cb;gv86,456]]]
```

Figure 8.2
Flowchart—procedure for the allocation of white space


c9
Art already on next page ?
No gv195 = 0
Compute space remaining in current text block
Will art fit current page ?
Yes gv90 ⩾ gv6
Compute ½ current text block depth
Yes gv195 = 1
No gv90 < gv6
Will art fit on next page ?
Yes gv86 ⩾ gv6
Is art already at top of next page ?
Yes gv143 ≠ 24
Compute ½ remaining text column (next page)
Art less than ½ nominal text column ?
Yes gv6 < gv92
Art less than ½ current text column ?
Yes gv6 < gv91
Adjust text block origin (current page)
No
Adjust text block origin (next page)
No
No* (*Space will not be allocated; that is, it is "lost" and the citation must be moved)
No*
Art less than ½ remaining text column ?
Yes gv6 < gv91
Adjust text block bottom boundary (current page)
Adjust text block origin (next page)
No
Adjust text block bottom boundary (next page)
Set carryover indicator to 1
Adjust text block bottom boundary (next page)
End

Insufficient space on the current page

If **gv90** is less than **gv6**, there is insufficient space remaining on the present page to accommodate the white space and it must be allocated on the following page. It is not necessary, in this case, to compute one-half column depth—no white space has yet been allocated on the following page and no text has been set on the following page. The half-column depth can be stated explicitly in the conditional procedure. If **gv6** is less than 228 points (one-half nominal text column depth) the white space will be allocated at the top of the text column by moving the text block origin downward and shortening the text block bottom boundary by the value assigned to **gv6** as in the earlier case. In this case, the new block origin and the new text block bottom boundary will be computed as before and assigned to general variables in order that they can be assigned during the next execution of the *when bottom*

Figure 8.3 Format statement for the allocation of white space in single column text

```
c9 ≡ [eq,gv195,1[[ge,gv86,gv6[[ne,gv143,24[[qo,gv91,gv86,2;lt,gv6,gv91[[ad,gv143,gv143,gv6]];df,gv86,gv86,gv6]]]]]];
     eq,gv195,0[[eq,cy,0[[eq,cx,0[[gv90,0]];ne,cx,0[[gv90,ps]]]];
                ne,cy,0[[eq,cx,0[[df,gv90,cy,bl;ad,gv90,gv90,ps]];ne,cx,0[[ad,gv90,cy,ps]]]];df,gv90,bb,gv90;
                ge,gv90,gv6[[qo,gv91,bb,2;lt,gv6,gv91[[ad,gv144,gv144,gv6;by,1,gv144]];df,bb,bb,gv6]];
                lt,gv90,gv6[[lt,gv6,gv92[[ad,gv143,gv6,24]];df,gv86,gv86,gv6;gv195,1]]]]]
```

boundary procedure.

The value of the block origin is computed as the sum of **24** (the nominal text block origin) and **gv6**, the new value is assigned to **gv143**. The bottom boundary is the difference between **gv86** (**gv86** is used in the *when bottom* boundary procedure to set the text block bottom boundary) and **gv6**; the new value of the bottom boundary (next page) is assigned to **gv86**. Finally, the "art on next page" indicator is set equal to one. See line five of Figure 8.3.

Art already assigned to the next page

Assume now that the procedure just developed is executed and results in white space being allocated on the following page and further that another citation for white space is encountered before the present page is closed. General variable 195 has the value one. (Line one of Figure 8.3.) If the white space being requested with this new citation is to be accommodated, two conditions must be true: there must be more space available on the next page than has been requested and the white space already on the next page must be at the top of the column. If either of these conditions is not true, the white space will not be provided for. If the incidence of requests for white space is high, the placement of citations in the input (ie., on the manuscript) requires care in order to avoid losing the white space.

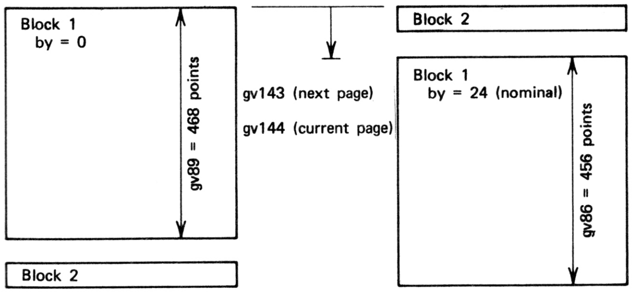
The two conditionals are nested: first, **gv86** (the bottom boundary of the next text column) must be greater than or equal to **gv6** and second, **gv143** (the Y-coordinate of the text block on next page) must be not equal to **24**—that is, the white space allocated in the next page is at the top of the text column. See line one of Figure 8.3.

If both of these conditions are true, the half-column depth is computed by dividing **gv86** by two and assigning the result to **gv91.** If the space requested (**gv6**) is less than **gv91** the Y-coordinate of the text block origin (**gv143**) is increased by **gv6**. (This action allocates the new space at the top of the next page with the space already allocated there.) Again only one conditional procedure is necessary because if **gv6** is greater than **gv91**, the requested space is allocated at the bottom of the text column—the block origin is not altered. In either event, **gv86** is decreased by **gv6**.

Modifications to the pagination algorithm

The pagination algorithm as it appears in Figure 8.1 has already been modified such that the text block bottom boundary for the new page being initialized is stated as **gv86**. It will now be further modified to state the Y-coordinate of the text block origin (in case white space has been carried forward) and to reinitialize the values of the general variables used in the allocation of white space. A block map of the chapter opening page and a continuation page, showing the general variables in question, is given in Figure 8.4 and the

Figure 8.4
Block map—single column text



Notes:

1. Depth of text is 468 points and 456 points on the chapter opening and continuation pages, respectively. At the start of the format statement, gv89 and gv86 are therefore initialized to 468 and 456 points respectively.

2. If white space is allocated on the first continuation page, during composition of the chapter opening page, gv86 will be decremented by the amount of that white space.

3. During execution of the *at bottom* procedure (at the close of the chapter opening page), gv89 is reset to 456 points, which is the nominal bottom boundary for all remaining pages in the chapter, and—after the continuation page text block bottom boundary has been set to the value stored in gv86—gv86 is reset to 456 points.

4. The Y-coordinate of the text block origin is zero and 24 points on the chapter opening and continuation pages, respectively. At the start of the format statement, gv144 and gv143 are therefore initialized to zero and 24 points, respectively. General variable 144 is used for the current page and gv143 is used for the next page.

5. If white space is allocated at the top of the first continuation page, during composition of the chapter opening page, gv143 will be decremented by the amount of that white space.

6. During execution of the *at bottom* procedure (at the close of the chapter opening page), gv144 is reset to 24 points. After the continuation page text block origin has been set equal to the value stored in gv143, gv143 is also reset to 24 points.

finished pagination algorithm, relabeled **c5** is given in Figure 8.5; the additions are set in boldface type.

The "art on the next page" indicator (**gv195**) is initialized to zero in line two of Figure 8.4; at chapter opening, no white space has yet been allocated to the following page. If white space has been allocated to the top of the page being initialized, during composition of the page just completed, the value for the Y-coordinate of text block origin was assigned to **gv143**. The Y-coordinate of text block origin is therefore assigned the value stored in **gv143** in both bottom boundary procedures: lines 9 and 16 in Figure 8.5. The value of **gv144** (Y-coordinate of text block origin for the page being composed) is reinitialized to the nominal value as a precaution in the event that white space has not been carried forward to the page being initialized. These assignments also appear in lines 9 and 16. These assignments might be made in a conditional procedure based on **gv195** having the value zero; but reinitialization on each page gives the same result and requires less code.

Subsequently, in lines 13 and 20, **gv144** is assigned the value stored in **gv143** (and **gv195** is set back to zero) in a conditional procedure based on **gv195** having the value one. That is, if white space is carried to the page being initialized, the Y-coordinate of text block origin has been set equal to **gv143**, and now **gv144** is set equal to **gv143** in the event of further change as the current page is composed. Finally, in lines 15 and 22, the values of **gv86** and **gv143** (bottom boundary and Y-coordinate of text block origin for the next page respectively) are reinitialized to 456 and 24 points respectively. These are the nominal text block position and depth and, if white space is not carried to the following page during composition of the page being initialized by this execution of the bottom boundary procedure, they will still have these nominal values at the next execution of the bottom boundary procedure.

The codes to initialize **gv143** and **gv144** to 24 and zero respectively and to initialize **gv86** to 456 are added to the opening of the algorithm (lines 3 and 2). General variables 143 and 86 pertain to the following page, which is a continuation page. General variable 144, however, refers to the chapter opening (the current) page. The nominal value for Y-coordinate of text block origin on the chapter opening page is zero so this initialization is consistent. Recognize, however, the undesirability of white space being allocated to the top of a chapter opening page. The algorithms, as developed, will allocate white space to the top of the opening page (above the chapter headings) if that circumstance is encountered. If a citation which specifies a small depth of white space is encountered in the opening lines of a chapter, the procedure will allocate the space to the top of the text column.

There are several ways to evade this possibility: by putting the chapter headings in a separate block and moving the opening page text block downward (and shortening its bottom boundary) or by modifying the white space procedure such that a request for white space encountered on a chapter opening page is always satisfied by allocating the space at the bottom of the opening text column or on the following page. From the practical viewpoint it seems as well to avoid the circumstance in markup or to move the citation if the circumstance is encountered.

WHITE SPACE IN TWO COLUMN TEXT

THE ALGORITHM for two-column text with footnotes developed in Chapter Seven will be modified to accommodate the allocation and placement of white space and a procedure to position white space will be developed. As in the single column case, the procedure will allocate white space by adjusting text block origins and bottom boundaries. The rules for placement are:

- If the white space to be provided will fit in the column

Figure 8.5
Format statement for single column text with footnotes and space allocation capability

```
c5 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,470;bx,0,12;bx,3,12;
      pd,468;gv89,468;gv86,456;gv194,0;gv195,0;
      gv182,0;gv143,24;gv144,0;ad,pn,pn,1;fn,1;

      sb,1;lb,0;tb,0;rb,288;bb,468;ju;ib;

      sb,2;lb,0;tb,0;rb,288;bb,468;
      ps,10;bl,10;tf,6;ce;dc,pn;nl;

      sb,3;ju;wb[[gv192,cy;sb,0;bb,gv86;gv182,1]];cb;

      ab[[gv89,456;pd,480;jv;pg;sb,1;
         by,1,gv143;bb,gv86;gv194,0;gv144,24;
         eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                  ad,gv194,gv193,2;ad,gv90,gv193,2;
                  df,bb,gv89,gv90;gv182,0]];
         eq,gv195,1[[gv144,gv143;gv195,0]];
         ad,pn,pn,1;ib;sb,2;ps,10;tf,6;bb,gv86;
         Y2;nl;sb,3;ju;cb;gv86,456;gv143,24;by,2,0]];
      wb[[jv;pg;sb,1;by,1,gv143;bb,gv86;gv194,0;gv144,24;
          eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                   ad,gv194,gv193,2;ad,gv90,gv90,2;
                   df,bb,gv89,gv90;gv182,0]];
          eq,gv195,1[[gv144,gv143;gv195,0]];
          ad,pn,pn,1;ib;sb,2;ps,10;tf,6;bb,gv86;
          Y2;nl;sb,3;ju;cb;gv86,456;gv143,0]]]
```

being composed when the citation is encountered, the allocation shall be made in the current column;

- If the white space to be provided is less than one-half of nominal text column depth, the white space shall be provided at the top of the text column;
- If the white space to be provided is more than one-half of nominal text column depth, the white space shall be provided at the bottom of the text column;
- If there is insufficient space remaining in the column being composed to accomodate the requirement, the allocation shall be made in the next column—that is, if the citation is encountered in column one of the page being composed and there is not sufficient space in column one, the allocation shall be made in column two;
- The procedure shall be capable of deferring white space within two page sets—that is, the procedure shall provide white space at the first available column top or bottom in the current or succeeding page.

A procedure will be developed to make the decisions stated in the rules for placement and to compute new block origins and bottom boundaries necessary to provide the specified white space. If the white space fits on the citation page, the new block origin will be stated within the procedure; if it fits within the column in which the citation appears, the new text block bottom boundary will also be stated within the procedure. If the white space is deferred to a succeeding column, the value of the bottom boundary will be stored in a general variable, and if it is deferred to the top of column on the next page, the value of the block origin will be stored in a general variable for use when the bottom boundary is executed.

The algorithm for two column text with footnotes developed in Chapter Seven (Figure 7.14) is reproduced in Figure 8.6. Note that statement of the text block bottom boundary has been changed to **gv99** for block four (in line eight of Figure 8.6) and to **gv168** for block one (line eleven). Further, the recalculation of the block four bottom boundary in the event of footnote carry from column one to column two (line nine of the figure) is in terms of **gv99** rather than an explicit 550 points. Finally, the general variables 98, 99, 168, and 169 are initialized to 550 points in the third line of the format statement. These changes are set in boldface type in Figure 8.6.

The values of these general variables will be manipulated in the white space procedure (as will the values of the text block origins). The bottom boundaries for columns one and two of the current page are **gv98** and **gv99** respectively. For columns one and two on the next page, the bottom boundaries are **gv168** and **gv169** respectively. The pagination algorithm of Figure 8.6 will be further modified to include statement of block origins and other housekeeping activities subsequently.

The white space procedure

The procedure for placing white space will, as in the single column case, comprise a number of conditional procedures to comply with the stated style rules. The first level procedures ask the question: Is white space already provided on the next page? General variable 200 will be used as the switch; if **gv200** is equal to zero, white space has not yet been assigned to the next page. If white space is assigned to the next page, **gv200** will be set to one.

The second level procedures ask the questions: has white space been assigned to the bottom of column two? at the bottom of column one? in that order. General variables 195, 196, 197, and 198 are used as "art at bottom" of columns one and two current page and columns one and two next page respectively. At the time that white space is allocated at the bottom of any column. The appropriate general variable is set to one. When the page is advanced, the values of these general variables must be appropriately set and reset to reflect conditions as the new page is initialized; for example, after the new page is started, what was "next page, column two" becomes "current page, column two". These and other housekeeping matters will be considered subsequently.

Art not yet assigned to the next page

The procedure for allocation and positioning white space is shown in figure 8.7. Assume that a "graphic

Figure 8.6
Format statements for two column text with footnotes

```
A7 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,268;bx,3,12;bx,4,204;
     by,4,0;bx,5,204;gv1,0;gv3,3;pd,550;ad,pn,pn,1;fn,1;
     gv98,550;gv99,550;

     sb,1;lb,0;tb,0;rb,180;bb,550;ps,10;bl,12;tf,4;juuib;

     sb,2;lb,0;rb,372;bb,550;ps,8;bl,10;tf,8;Y2;nl;

     sb,3;lb,0;rb,180;ju;
     wb[[gv192,cy;bb,gv88;gv182,1;eq,gv1,0[[sb,5]]]];cb;

     wb[[eq,gv1,0[[jv;sb,4;bb,gv99;eq,gv182,0[[ib;sb,5;cb]];
                 eq,gv182,1[[df,bb,gv99,gv91;gv182,0]];
                 gv3,5;gv1,2]];
        eq,gv1,1[[jv;pg;sb,1;bb,gv168;ad,pn,pn,1;ib;sb,2;
                 lb,0;rb,372;ps,8;tf,8;Y2;nl;sb,3;lb,0;
                 rb,180;ps,8;cb;gv3,3]];
        df,gv1,gv1,1]]]
```

citation" is to specify ten lines of white space and that text is being set 10/12. The citation is **[gv6,120;b8]** in which **gv6** is set equal to 120 points and **b8** is the format label given the white space procedure. Assume that **gv200** has the value zero indicating that no white space has yet been assigned to the next page. Two conditional procedures based on the value of **gv196** ask the question: has white space been assigned to the bottom of column two of the current page? these procedures appear in lines 13 and 15 of Figure 8.7.

Assume first that **gv196** has the value one. If so, the request for white space must be assigned to column one on the next page. The fact that no space is yet allocated in column one on the next page derives from the fact that **gv200** is assumed to have the value zero. Nested within the **gv196** equal to one conditional procedure are two further procedures based on the relationship between the space requested and the half-nominal column depth. At the end of line 3, if the space requested is greater than half-column, **gv197** is set equal to one because the allocation will be made at the bottom of the column. At the beginning of line 14, if the space requested is less than half-column, the value of **gv143** is increased by the value assigned to **gv6**. General variable 143 is used to store the Y-coordinate of block one on the next page; the allocation is to be made at the top of column one on the next page. Finally, at the end of line 14, **gv168** is decreased by the value assigned to **gv6** and **gv200** is set to one. General variable 168 is used to store the bottom boundary for block one on the next page and is changed whether white space is assigned to the top or the bottom of column one. General variable 200 is set to one in either case to indicate that white space has been assigned on the next page.

Assume now that **gv196** has the value zero. The request may be satisfied on the current page. The amount of uncommitted space in the text column being composed is computed in lines 15 and 16 of Figure 8.6. This done, the identity of the current text column is determined by conditional procedures based on the value of **gv1**, the column indicator used in the bottom boundary procedure. If the value of **gv1** is zero, (line 17 of the figure) indicating that the current text column is one, two conditional procedures based on the value of **gv195** ask the question: has white space been assigned to the bottom of this column? These two procedures appear in lines 17 and 21 of Figure 8.6.

Consider first the event that art has already been assigned to the bottom of column one and **gv195** has

Figure 8.7
Procedure for the allocation of white space in two column text

```
B8 ≡ [eq,gv200,1[[eq,gv198,0[[eq,gv197,1[[ge,gv169,gv6[[ge,gv6,gv92[[gv198,1]];
                                                                     lt,gv6,gv92[[ne,gv149,gv123[[gv198,1]];
                                                                                    eq,gv149,gv123[[ad,gv149,gv149,gv6]]]];df,gv169,gv169,gv6]]]];
                                     eq,gv197,0[[ge,gv168,gv6[[ge,gv6,gv92[[gv197,1]];
                                                                     lt,gv6,gv92[[ne,gv143,gv123[[gv197,1]];
                                                                                    eq,gv143,gv123[[ad,gv143,gv143,gv6]]]];df,gv168,gv168,gv6]];
                                                    lt,gv168,gv6[[ge,gv169,gv6[[ge,gv6,gv92[[gv198,1]];
                                                                                     lt,gv6,gv92[[ne,gv149,gv123[[gv198,1]];
                                                                                                    eq,gv149,gv123[[ad,gv149,gv149,gv6]]]];df,gv169,gv169,gv6]]]]]]]]];

         eq,gv200,0[[eq,gv196,1[[ge,gv6,gv92[[gv197,1]];
                                     lt,gv6,gv92[[ad,gv143,gv143,gv6]];df,gv168,gv168,gv6;gv200,1]];
                       eq,gv196,0[[eq,cy,0[[gv90,0;ne,cx,0[[gv90,ps]]]];ne,cy,0[[eq,cx,0[[df,gv90,cy,bl;ad,gv90,gv90,ps]];ne,cx,0[[ad,gv90,cy,ps]]]];df,gv90,bb,gv90;
                                     eq,gv1,0[[eq,gv195,1[[ge,gv6,gv92[[gv196,1]];
                                                               lt,gv6,gv92[[ne,gv150,gv123[[gv196,1]];
                                                                              eq,gv150,gv123[[ad,gv150,gv150,gv6;by,4,gv150]]]];df,gv99,gv99,gv6]];
                                                 eq,gv195,0[[ge,gv90,gv6[[ge,gv6,gv92[[gv195,1]];
                                                                               lt,gv6,gv92[[ne,gv144,gv123[[gv195,1]];
                                                                                              eq,gv144,gv123[[ad,gv144,gv144,gv6;by,1,gv144]];df,bb,bb,gv6;gv98,bb]]]];
                                                               lt,gv90,gv6[[ge,gv6,gv92[[gv196,1]];
                                                                              lt,gv6,gv92[[ad,gv150,gv150,gv6;by,4,gv150]];df,gv99,gv99,gv6]]]]]];
                                     eq,gv1,1[[ge,gv90,gv6[[ge,gv6,gv92[[gv196,1]];
                                                                lt,gv6,gv92[[ne,gv150,gv123[[gv196,1]];
                                                                               eq,gv150,gv123[[ad,gv150,gv150,gv6;by,4,gv150]]]];df,bb,bb,gv6;gv99,bb]];
                                                 lt,gv90,gv6[[ge,gv6,gv92[[gv197,1]];
                                                                lt,gv6,gv92[[ad,gv143,gv143,gv6]];df,gv168,gv168,gv6;gv200,1]]]]]]]]
```

the value one. The present request will be satisfied in column two. It is not necessary to query whether the art will fit in column two since no art has yet been assigned to the bottom of column two. If the space requested is greater than one-half column, **gv196** is set equal to one (at the end of line 17) since the allocation will be made at the bottom of column two. If the space requested is less than one-half column, the first choice is that it be placed at the top of column two. But the possibility exists that art has already been placed there. Two conditional procedures based on the value of **gv150** determine the proper course. If **gv150** is not equal to **gv123**, art has already been assigned to the top of column two and **gv196** is set equal to one since the present request will be placed at the bottom of column two (end of line 17). If **gv150** is equal to **gv123** (line 19), no space has yet been assigned to the top of column two and the present request can be satisfied by placing it at the top of column two. This is accomplished by incrementing the value of **gv150** and then setting the Y-coordinate of block four equal to **gv150.** Finally the value of **gv99** is decremented by the value assigned to **gv6**. The new value for the column two bottom boundary is carried in **gv99** and will be assigned at the next execution of the bottom boundary procedure.

Consider now the event that art has not been assigned to the bottom of column one and **gv195** has the value zero (line 21 of the figure). Two conditional procedures (in lines 21 and 23 of the figure) based on the value of **gv90** and **gv6** determine the course of action. General variable 90 was, in line 16 of the figure, assigned a value equal to the space remaining in the current column. If the value of **gv90** is greater than that of **gv6** and if the value of **gv6** is greater than **gv92**; that is, the request is greater than one-half column, **gv195** is assigned the value one since the requested space will be assigned to the bottom of column one (line 21). If the value of **gv90** is greater than that of **gv6** , if the value of **gv 6** is less than that of **gv92**, and if **gv144** is not equal to **gv123** ; that is, the request is less than one-half column but art has already been placed at the top of the column, **gv195** is assigned the value once since the space will be assigned to the bottom of column one (line 21). If art has not been assigned to the top of column one, **gv144** and **gv123** are equal; the value of **gv144** is increased by the value of **gv6** and the Y-coordinate of block one is set equal to the resulting value (line 23). Finally the bottom boundary of block one is decremented by the value of **gv6** and the new value of the bottom boundary is assigned to **gv98**.

If the value of **gv90** is less than that of **gv6** the request cannot be satisfied in column one. The assignment will be made in column two; at the bottom if the space requested is greater than one-half column (line 25) and at the top if the request is less than one-half column. If the assignment is made at the bottom of column one, **gv196** is set equal to one. If the assignment is made at the top of column two, the block origin of block four is moved. In either case, a new value for the block four bottom boundary is computed by decrementing **gv99** by the value assigned to **gv6**. If the value of **gv1** is one, the citation has been encountered in column two; the amount of space remaining in the current column, computed earlier and assigned to **gv90**, pertains to column two. Again, two conditionals in lines 28 and 30 test the relationship between **gv90** and **gv6** to determine whether or not the request can be accommodated in the current column, and in fact, on the current page. Lines 28 through 31 are completely analogous to lines 16, 17, and 18. If the space is to be placed at the bottom of column two (because the request is greater than one-half column or if it is less than one-half column but art already has been placed at the top of the column), **gv196** is assigned the value one (lines 28 and 29). If the space is to be placed at the top of column two, the value of **gv150** is adjusted and the Y-coordinate of block four is set equal to the resulting value. Finally, the value of the column two bottom boundary is decremented by the amount of the space requested and this new bottom boundary value is assigned to **gv99**. If the current request cannot be accommodated in column two, it will be assigned to column one of the next page. Lines 32 and 33 are analagous to lines 13 and 14. If the request is for more than one-half column, it is placed at the bottom and **gv197** is assigned the value one. If for less than one-half column, the space is placed at the top of the column by adjusting the value of **gv143.** Finally the value of **gv168** is decremented and **gv200** is assigned the value one. The origin and bottom boundary of block one will be assigned during the next execution of the bottom boundary procedure.

Art already assigned to the next page

If the procedure just developed were executed and the request was satisfied by assigning the space to the next page and if another citation is encountered before the current page is closed, **gv200** will have the value one. If the earlier request was satisfied by an assignment at the bottom of column two of the next page, this latest request cannot be accommodated and will be lost. In order for the latest request to be placed, **gv198** must have the value zero (line one of the figure).

If art has already been placed at the bottom of column one on the next page, **gv197** has the value one and **gv169** which holds the value of the block two bottom boundary (next page), must be greater than **gv6** if the current request is to be accommodated. That is, there must be more space remaining in column two-next page than the current request in order for the request to be accommodated. If there is not, the space will be lost. If the placement is to be at the bottom of column two (because it is for more than one-half column or if it is for less than one-half column but art has already been placed at the top of column two), **gv198** is assigned the value one. If the request is for less than one-half column and art has not yet been assigned to the toop of column two, **gv149** is incremented by the value assigned to **gv6**. Finally, **gv169** is decremented by the amount of the request.

If no space has been reserved at the bottom of column one-next page, **gv197** has the value zero and two conditional procedures test the value of **gv168** to determine whether the requested space will fit in column one. Assuming that it will, it goes to the bottom if it is greater than one-half column or if space has already been reserved at the top of column one. In either event **gv197** is assigned the value one (lines five and four). If the request is less than one-half column and space has not yet been assigned to the top of the column, the value of **gv143** is increased. Finally, **gv168** is decremented by the amount of the request (line eight).

If the request cannot be satisfied in column one-because there is insufficient space remaining to accommodate it, the assignment will be made in column two. If it is placed at the bottom of column two, **gv198** is assigned the value one (lines nine and ten). If it is placed at the top, the value of **gv149** is decremented (line eleven) and finally **gv169** is decremented appropriately.

The procedure has been considered out of sequence because it is more natural to take it that way. In execution, the "art already assigned to the next page" portion must come first.

Modifications to the pagination algorithm

The pagination algorithm as it appears in Figure 8.6 has already been modified such that the bottom boundary of the text block being initialized in a given execution of the bottom boundary is stated as **gv99** for block four and **gv168** for block one. As the page changes, the "next page bottom boundaries"—which were computed in **B8** if white space was assigned to the next page—become the current page bottom boundaries. The general variables which represent the current page text block bottom boundaries must be initialized; at the start of line 15 of Figure 8.8, **gv98** and **gv99** are set equal to **gv168** and **gv169** respectively. In line sixteen **gv168** and **gv169** are then re-initialized to the nominal 550 points.

The block origins and the associated general variables are subject to the same operations. In lines 13 and 14, the Y-coordinates of blocks one and four are set equal to the values computed and stored in general variables 143 and 149 respectively. Subsequently, in line fifteen, general variables 144 and 150, which represent the block origins for the current page, are updated to the values carried forward from the preceding page in general variables 143 and 149. Then, in line seventeen, general variables 143 and 149 are re-initialized to zero.

The "art at bottom of column indicators" are treated similarly. General variables 195 and 196, are initialized to the values of general variables 197 and 198 (line 16) and general variables 197, 198, and 200 are than set to zero in line 17. These operations all appear in boldface type in Figure 8.8.

Figure 8.8
Format statements for two column text
with footnotes and white space capability

```
A7 ≡ [bx,1,12;by,1,0;bx,2,12;by,2,268;bx,3,12;bx,4,204;
     by,4,0;bx,5,204;gv1,0;gv3,3;pd,550;ad,pn,pn,1;fn,1;
     gv143,0;gv149,0;gv98,550;gv99,550;gv168,550;
     gv169,550;gv195,0;gv196,0;gv197,0;gv198,0;gv200,0;

     sb,1;lb,0;tb,0;rb,180;bb,gv98;ps,10;bl,12;tf,4;ju;ib;

     sb,2;lb,0;rb,372;bb,550;ps,8;bl,10;tf,8;Y2;nl;

     sb,3;lb,0;rb,180;ju;
     wb[[gv192,cy;bb,gv88;gv182,1;eq,gv1,0[[sb,5]]]];cb;

     wb]]eq,gv1,0[[jv;sb,4;bb,gv99;eq,gv182,0[[ib;sb,5;cb]];
                    gv182,0;gv3,5;gv1,2]];
        eq,gv1,1[[jv;pg;sb,1;bb,gv168;ad,pn,pn,1;ib;
                    sb,2;lb,0;rb,372;ps,8;tf,8;Y2;nl;sb,3;lb,0;
                    rb,180;ps,8;cb;by,1,gv143;
                    by,4,gv149;gv3,3]];
        gv98,gv168;gv99,gv169;gv144,gv143;gv150,gv149;
        gv195,gv197;gv196,gv198;gv168,550;gv169,550;
        gv143,0;gv149,0;gv197,0;gv198,0;gv200,0;
        df,gv1,gv1,1]]]
```

PART IV

THE FORMAT MANUAL CONCEPT

Part IV introduces the concept of generalized format statements in which the details of style are specified as *general variables.* Generalization makes possible the application of standard algorithms to a wide range of titles for which the rules are the same but such details as column width, spacing, font, and sizes vary from one application to another.

Chapter Nine presents the format statements for single-column text and footnotes from Chapter Seven and the subsidiary format statements of Chapter Six in their generalized form. The job specification necessary for using generalized format statements is introduced and subsequently a *style generator* which relates the concept to the traditional style specification is developed. The two-column text, with footnotes, format statements are given at the close of the chapter.

Chapter Ten is an example of the application of the generalized format statements to specific titles. The books to be considered are introduced by their style specifications; the job specifications are developed, manuscript mark-up is considered, and sample pages are displayed.

The format statements developed in Chapters Five through Eight, generalized in accordance with the style generator scheme presented in chapters Nine and Ten, are set down together in Appendix B as a PAGE–1 Format Manual. They reflect the style rules followed in the developments of Part III.

The practice of typography presents many circumstances in which style decisions must be made and users are likely to have different opinions about the alternatives to be employed at decision points. Appendix B represents only one set of solutions to a rather limited set of typographic requirements. The point is that each user must make a comprehensive statement of his style requirements—a statement that includes all of the alternatives available at each decision point. Then the work of the composition programmer may begin.

9

GENERALIZED FORMAT STATEMENTS

AS WE HAVE SEEN, a format statement is a sequence of Page–1 codes which when executed by the processor will direct composition of the input manuscript according to the rules on which the format statement is based. The theory behind generalized format statements is that many books are composed in keeping with the same general rules—only the details of spacing, column width, font and size, and others change from title to title or series. Figure 9.1 shows the spatial relationships which exist among the blocks in which folios and running heads, primary text, and footnotes have been set in the developments of Chapters 5 through 8. The locations of these blocks on a page were specified by constant values given in the style specification segments quoted in the several developments.

General variables to specify typographic detail

In the format statement for single column text with footnotes—final form given in Figure 7.10—the Y-coordinate of block 2 was specified as 470 points; the analogous dimension in Figure 9.1 is specified to be the value assigned to **gv121**. The bottom boundary of the text block on the chapter opening page was

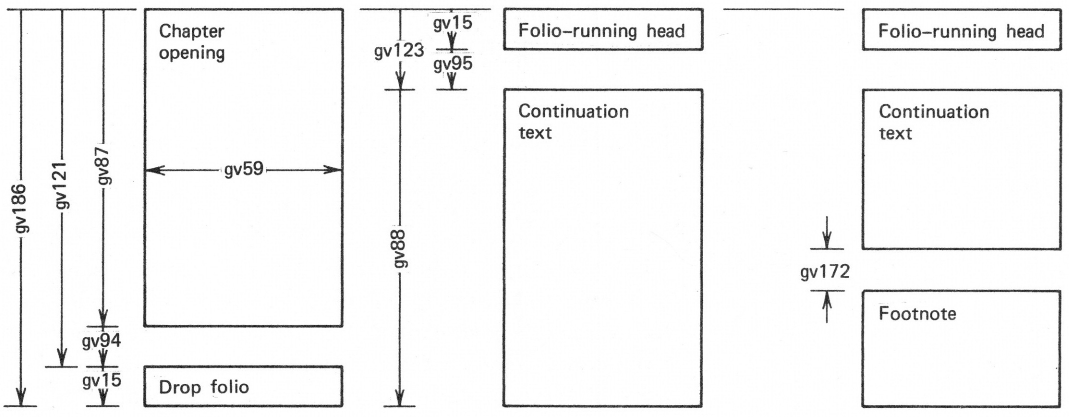

gv186
gv121
gv87
Chapter opening
gv59
gv94
gv15
Drop folio
gv123
gv15
gv95
gv88
Folio–running head
Continuation text
gv172
Folio–running head
Continuation text
Footnote

Figure 9.1
Block relationships

468 points; now it is the value assigned to **gv87** or simply **gv87**.

For the continuation page, the format statement of Figure 7.10 provides fourteen points between running head and text by setting the value of the Y-coordinate of block one origin to 24 points. The block one Y-coordinate in Figure 9.1 is **gv123**; it is also the sum of **gv15** (running head point size) and **gv95** (sink of text below the running head).

Measure was specified as 288 points (24 picas); in Figure 9.1 it is **gv59**. The space between text and footnotes is not generally given in a style specification; rather it is left to the judgement of the person in the composing room who makes the pages. The space is considered variable and is used to effect vertical justification of the page. The algorithm must be given a minimum space assignment. In Figure 7.10 it is two points; in Figure 9.1 it is **gv172.**

Most of the spacing details which must be specified are given in Figure 9.1; missing are the spacing around side heads, chapter heads, and extract material. These will be considered later. The remaining details of the single column text format statement involve fonts and sizes; these are set down in Figure 9.2. No body lead is quoted for the folio/running head because the running head is almost always only one line; the body lead is not used. Nor is body lead necessary for the footnote citation.

Generalized format statements for single-column text with footnotes

A straight substitution of the general variables noted on Figure 9.1 and those tabulated in Figure 9.2 into the format statement of Figure 7.10 leads to the format statements in Figure 9.3.

Before making a similar substitution in the format statements for footnotes, two more distances must be specified: the amount of the *baseline up* commands prior to the typesetting of the footnote citations. The amount of the "up" in the footnote block is always the same: 32 times the difference between the footnote and the citation point sizes. In the example of Chapter 7, the point sizes were eight and five respectively, the difference is three points, and the "up" is 96 units. The same rationale follows in the text block; but the citation may appear in primary or secondary text, or in any of several levels of side heads. Other problems may occur if it appears in a primary chapter head—where the head is in a much larger point size—and the single point size specified for both text and footnote block citation size doesn't work. If one is willing to stipulate that a raised five point character is a reasonable size for the footnote citation over a range of eight to twelve point characters in which primary text, extracts, and footnotes might be set, the following method of computing the amount of the "up" is sufficient. Subtract the citation point size from the "current" point size and multiply the difference by 32; that is, subtract **gv25** from **ps** and multiply by 32. The format statements to start and close footnotes also appear in Figure 9.3.

Figure 9.2
Fonts and sizes for generalized format statements

font requirement	*point size*	*body lead*	*typeface*
Primary text	gv11	gv26	gv41
Folio/running head	gv15	-	gv45
Footnote	gv14	gv29	gv44
Footnote citation	gv25	-	gv55

Figure 9.3
Generalized format statements for single-column text and footnotes

```
C1 ≡ [bx,1,gv122;by,1,0;bx,2,gv122;by,2,gv121;
       bx,0,gv122;bx,3,gv122;pd,gv87;gv89,gv87;
       gv194,0;gv182,0;ad,pn,pn,1;fn,1;

       sb,1;lb,0;tb,0;rb,gv59;bb,gv87;ju;ib;

       sb,2;lb,0;tb,0;rb,gv59;bb,gv87;ps,gv15;bl,ps;tf,gv45;Y4;nl;

       sb,3;bl,gv29;ju;wb[[Y3]];cb;

       ab[[gv89,gv88;pd,gv186;Y1;by,1,gv123;by,2,0]];
       wb[[Y1]]]

C2 ≡ [jv;pg;sb,1;bb,gv89;gv194,0;
       eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                   ad,gv194,gv193,gv173;ad,gv90,gv193,gv172;
                   df,bb,gv89,gv90;gv182,0]];
       ad,pn,pn,1;ib;sb,2;ps,gv15;tf,gv45;bb,gv89;Y2;nl;
       sb,3;bl,gv29;ju;cb]

C3 ≡ [gv192,cy;sb,0;bb,gv89;gv182,1]

[sy,Y1[[C2]];sy,Y2[[G1]];sy,Y3[[C3]];sy,Y4[[G6]]]

F3 ≡ [ad,gv90,cy,ps;ad,gv90,gv90,gv172;
       ad,gv90,gv90,gv194;
       ib;df,bb,gv89,gv90;ps,gv25;tf,gv55;
       up,gv174;dc,fn;up,0;ps,gv14;tf,gv44]

F4 ≡ [nl;eq,gv182,0[[df,gv192,cy,gv173]];
         eq,gv182,1[[df,gv193,cy,gv173]];cb;
       df,gv160,pd,gv192;by,3,gv160;ad,gv90,gv192,gv172;
       ne,gv194,0[[ad,gv90,gv90,gv194;
                   df,gv161,gv160,gv194;by,0,gv161]];
       df,bb,gv89,gv90;df,gv175,ps,gv25;ml,gv175,gv175,32;
       gv2,ps;gv4,tf;ps,gv25;tf,gv55;up,gv175;dc,fn;up,0;
       ps,gv2;tf,gv4;ad,fn,fn,1]
```

Job specification

Before the format statements of Figures 9.3 are used, the general variables which specify the detail must be given values. This is done with a Job Specification: a straightforward listing which assigns appropriate values to the general variables. The job specification must be typed into input ahead of the first appearance of a format label. The job specification for application of the format statements of Figure 9.3 to the specific requirements of the continuing example is given in Figure 9.4.

The style generator

Generation of the job specification of Figure 9.4 requires some computation by the person who applies the format statements to a typesetting problem and some knowledge of how the numbers are used in the algorithms. It is possible and perhaps desirable to write another format statement which accepts as inputs the things which appear in a conventional style specification and generates as output the values required by the format statements for text and footnotes.

Such a format statement would be executed once at the beginning of a job. The advantage of such a scheme is that the person who uses the format statements needs know little if anything about the Page–1 language.

The format statement to generate values for the format statements of Figure 9.3 from the information generally given in style specifications is shown in Figure 9.5. The depth of the nominal (ie., continuation page) text block is computed from the point size, leading, and the number of lines per page.

$$\text{Depth} = \mathbf{bl} \times (\mathbf{n} - \mathbf{1}) + \mathbf{ps}$$

in which **n** is the nominal number of lines per page and **ps** and **bl** are the point size and body lead for primary text.

The Y-coordinate of the text block on a continuation page is the sum of the point size for the folio/running head and the space between the running head and the first line of text. That is, **gv123** is computed as the sum of **gv15** and **gv95**. Depth of the type page is the sum of the Y-coordinate of the origin of the text block and its depth; **gv186** is set to the sum of **gv123** and **gv88**.

Figure 9.4
The job specification

```
[gv121,470;gv122,12;gv123,24;gv186,480;gv87,468;
 gv88,456;gv59,288;gv172,2;gv11,10;gv26,12;
 gv41,4;gv15,10;gv45,6;gv14,8;gv29,10;gv44,4;
 gv25,4;gv55,4;gv172,2;gv173,2;gv174,96]
```

Figure 9.5
The style generator

Nominal text block depth	[df,gv88,gv60,1; ml,gv88,gv88,gv26; ad,gv88,gv88,gv11;
Measure	ml,gv59,gv59,12;
Text block origin	ad,gv123,gv15,gv95;
Depth type page	ad,gv186,gv123,gv88;
Drop folio block origin	df,gv121,gv186,gv15;
Opening text block depth	df,gv87,gv121,gv94;
Footnote leading	df,gv173,gv29,gv14;
Up for footnote citation	df,gv174,gv14,gv25; ml,gv174,gv174,32]

The Y-coordinate of the drop folio block (block two on the chapter opening page) is computed by subtracting folio point size from depth of the type page. The depth of the text block on the chapter opening page is the difference between the Y-coordinate of the drop folio block and the space above the drop folio.

The last two values computed are the leading in the footnote block: the difference between point size and body lead; and the amount of the "up" required in the footnote block to simulate a superior digit for footnote citations.

Job specification if a style generator is used

The new job specification, which must precede the detail generator in input, is shown in Figure 9.6. The values specified are taken directly from the style specification; the user need not make any computations.

Figure 9.6
The new job specification

Nominal lines per page	[gv60,38;
Allowance for line numbers	gv122,12;
Measure	gv59,288;
Text-footnote space	gv172,2;
Space above drop folio	gv94,2;
Space below running head	gv95,14;
Font for text	gv11,10;gv26,12;gv41,4;
Font for folio/running head	gv15,10;gv45,6;
Font for footnotes	gv14,8;gv29,10;gv44;
Font for footnote citation	gv25,4;gv55,4]

Figure 9.7
Fonts, sizes, and spacing for generalized format statements

typographic format	*point size*	*body lead*	*typeface*	*space*	
				above	*below*
Primary chapter head	gv17	gv32	gv47	gv107	-
Secondary chapter head	gv18	gv33	gv48	gv108	gv110
First level side head	gv20	gv35	gv50	gv111	gv113
Second level side head	gv21	gv36	gv51	gv114	gv116
Secondary text	gv12	gv27	gv42	gv102	gv104

Generalizing the subsidiary format statements

The same approach may be used to generalize the format statements of Chapter Six for secondary text, side heads, and chapter heads. The general variables chosen to specify the fonts, sizes, and the spacing are set down in Figure 9.7.

A straightforward substitution of general variables into the format statements of Figure 6.8 (secondary text), of Figure 6.9 (first and second level side heads), and into the format statements which answer the problem at the end of Chapter Six (chapter heads and chapter opening) yield the format statements shown in Figures 9.8 through 9.10. The detail generator segments which compute the values required from style specification inputs are shown.

In Figure 9.8, the amount of the sink above secondary text is specified by **gv102** if **cx** is not zero and by **gv103** if **cx** is equal to zero. The user of the statement specifies the desired sink by assigning the number of points to **gv102** in his job specification. The detail generator then adjusts the value of **gv102** by adding primary text point size. This is because of the assumption that secondary text always follows primary text. The value of **gv103** is computed next by subtracting primary text body lead.

General variable 61 specifies the indention to be given secondary text; it is stated in ems. The generator next multiplies the input value of **gv61** by the secondary text point size (**gv12**) to change **gv61** to points so that it may be used to specify a left boundary for secondary text.

Finally values for **gv104** and **gv105** for the "return-to-primary-text" format statement. The computations are completely analageous to those for **gv102** and **gv103**.

The format statements for first and second level side heads and the accompanying detail generator are given without further comment in Figure 9.9.

The format statements for chapter heads and opening, Figure 9.10, show a straightforward substitution of general variables. The input values are adjusted as shown.

In practice, a book may require format statements for primary text, footnotes, folio and running heads, secondary text, and two levels of side heads. It is convenient to write a generator for a specific set of format statements which are regularly used together by stringing together the individual generators. Indeed, a style generator can accommodate a number of sets of similar kinds of format statements. At the same time it is worthwhile to add the X-coordinates of all of

Figure 9.8
Generalized format statements for secondary text

```
J1 ≡ [ne,cy,0[[eq,cx,0[[dj,gv103]];ne,cx,0[[dj,gv102]]]];
       eq,cy,0[[ne,cx,0[[dj,gv102]]]];
       ps,gv12;bl,gv27;tf,gv42;lb,gv61]

J2 ≡ [ne,cy,0[[eq,cx,0[[dj,gv105]];ne,cx,0[[dj,gv104]]]];
       eq,cy,0[[ne,cx,0[[dj,gv104]]]];
       ps,gv11;bl,gv26;tf,gv41;lb,0]

J3 ≡ [ne,cy,0[[eq,cx,0[[dj,gv101]];ne,cx,0[[dj,gv100]]]];
       eq,cy,0[[ne,cx,0[[dj,gv100]]]]]

[ad,gv102,gv102,gv11;df,gv103,gv102,gv26;ml,gv61,gv61,gv12;
 ad,gv104,gv104,gv12;df,gv105,gv104,gv27;]
```

Figure 9.9
Generalized format statements for side heads

```
H1 ≡ [ne,cy,0[[eq,cx,0[[dj,gv112]];ne,cx,0[[dj,gv111]]]];
       eq,cy,0[[ne,cx,0[[dj,gv111]]]];
       ps,gv20;bl,gv35;tf,gv50;lb,0;ql]

H2 ≡ [dn,gv113;ps,gv11;bl,gv26;tf,gv41;ju]

H3 ≡ [ne,cy,0[[eq,cx,0[[dj,gv115]];ne,cx,0[[dj,gv114]]]];
       eq,cy,0[[ne,cx,0[[dj,gv114]]]];
       ps,gv21;bl,gv36;tf,gv51;lb,0;ql]

H4 ≡ [dn,gv116;ps,gv11;bl,gv26;tf,gv41;ju]

[ad,gv111,gv111,gv11;df,gv112,gv111,gv26;ad,gv113,gv113,gv20;
 ad,gv114,gv114,gv11;df,gv115,gv114,gv26;ad,gv116,gv116,gv21]
```

the blocks to the generator. They never change during the job; it is sufficient to state their values once in the generator at the start of the job. Statement of the X-coordinates of the block origins can then be removed from the format statement for primary text. See the style generators given in the Format Manual, Appendix B.

Exercise: One might argue that a first (or second) level side head might follow secondary text as well as primary text and that the statements developed here will give different results in the two instances. Write a new statement similar to **H1** which computes the sink each time it is executed and provides the proper sink in both cases.

10

SAMPLE PAGES

THE FORMAT MANUAL, see Appendix B, will be used to demonstrate the use of generalized format statements. Throughout this chapter, reference should be made to Appendix B—not to earlier chapters in text—the generalized format statements in the format manual vary in detail from the developments given in chapters five through nine. Job specfications for two single-column books will be generated from the information given in their style specifications. Manuscripts showing the markup considerations and pages generated by a VideoComp will be displayed.

A non-fiction book

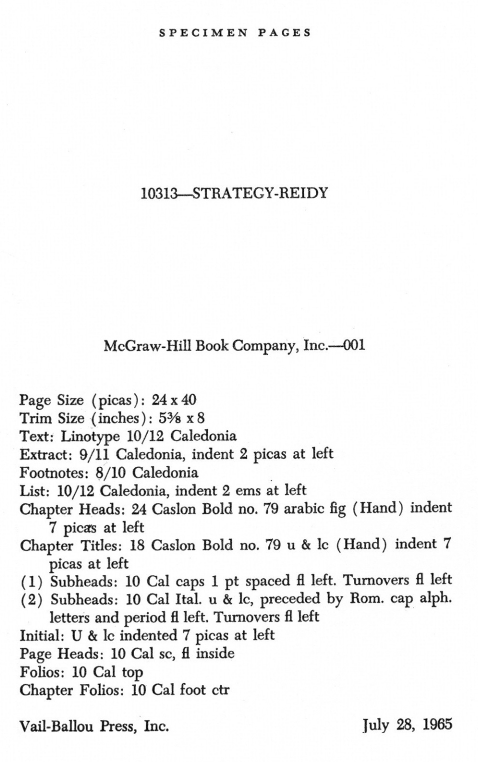
McGraw-Hill published *Strategy for the Americas,* by J.W. Reidy, in 1966. The style specification which governed the typesetting of the book is shown in Figure 10.1. A perusal of the style specification leads to a choice of **C1** for primary text (since the book has footnotes but no graphics) and **F3** and **F4** for footnotes. The book also requires two levels of side heads for which **H1** through **H4** will be used. The folio/running head treatment can be satisfied by **G1**, the centered drop folio by **G5** and the extract material by **J1** and **J2**. Chapter opening requirements will be satisfied by **D1** through **D3**. The font file is shown in Figure 10.2.

The job specification

The format manual lists the general variables which must be assigned values when a set of format statements is selected for use. The job specification for *Strategy for the Americas* is shown in Figure 10.3; an item by item comparison of Figures 10.1 and 10.3, with reference to the Format Manual as desired, will illustrate development of the job specification. Follow the sequence of items as they appear in Figure 10.3 and verify the value assigned by checking Figure 10.1:

- The job specification begins with an allow-

Figure 10.1
Style specification for
Strategy for the Americas



SPECIMEN PAGES

10313—STRATEGY-REIDY

McGraw-Hill Book Company, Inc.—001

Page Size (picas): 24 x 40
Trim Size (inches): 5⅜ x 8
Text: Linotype 10/12 Caledonia
Extract: 9/11 Caledonia, indent 2 picas at left
Footnotes: 8/10 Caledonia
List: 10/12 Caledonia, indent 2 ems at left
Chapter Heads: 24 Caslon Bold no. 79 arabic fig (Hand) indent 7 picas at left
Chapter Titles: 18 Caslon Bold no. 79 u & lc (Hand) indent 7 picas at left
(1) Subheads: 10 Cal caps 1 pt spaced fl left. Turnovers fl left
(2) Subheads: 10 Cal Ital. u & lc, preceded by Rom. cap alph. letters and period fl left. Turnovers fl left
Initial: U & lc indented 7 picas at left
Page Heads: 10 Cal sc, fl inside
Folios: 10 Cal top
Chapter Folios: 10 Cal foot ctr

Vail-Ballou Press, Inc. July 28, 1965

ance of twelve points for proof numbers;

- Column measure is 24 picas—288 points;
- A nominal page comprises 38 lines;
- The standard paragraph indention is one em;
- The typeface for the body of the book is Caledonia—the VideoFont equivalent is Video Gael—use typeface 18 which is Size Range II and includes point size ten;
- The body of the book is set 10/12;
- Secondary text is set 9/11 Caledonia (Video Gael, Size Range II) indented two ems (space above and below is not specified in the conventional style specification—four points above and below);
- Footnotes are set 8/10 Caledonia (Video Gael, Size Range II)—they are separated from primary text by six points (footnote citations are to be typeset using 5 point Video Gael and a raised baseline to simulate the superior number generally used. The text/footnote minimum space is not generally given in the conventional style specification);
- The folio is ten point Caledonia (Video Gael, Size Range II) and two points are specified above the chapter opening drop folio;
- The chapter number is specified to be 24 points Caslon Bold (Video Times Roman, Size Range III is substituted in this example) indented 7 picas and following a 2½ pica sink from the top of the page;
- The chapter title is specified to be 18 point Caslon Bold with 1½ picas between lines (Video Times Roman, Size Range III is substituted in this example—the 1 1/2 picas between lines is provided by 36 point body lead) indented 7 picas and having two picas between number and title;
- The chapter opening is indented 7 picas and follows a two pica sink below the last line of the chapter title;
- The first level side head is 10/12 Caledonia Bold (Video Gael Bold, Size Range II), with twelve points above and six points below, the side head is not indented but a standard paragraph indention follows the side head;
- The second level side head is 10/12 Caledonia Italic (Video Gael Italic, Size Range II), with ten points above and two points below, the side head is not indented but a standard paragraph indention follows the side head;
- The running heads are set in ten point Caledonia (Video Gael, Size Range II), with fourteen points below —the book title and the chapter title are used as right hand and left hand running heads respectively (only the left-hand running head definition is given in the job specification—the right-hand running head definition will be made at the beginning of each chapter);
- The text block and footnote block bottom boundary procedures for the footnote treatment selected are defined as **Y1** and **Y3** respectively and the folio/running head procedure and the drop folio procedure are defined as **Y2** and **Y4** respectively;
- In order to make use of true italic, the synonym names **Z8** and **Z9** are defined to be Video Gael Italic and Video Gael respectively; and finally,
- The style generator concludes the job specification.

Figure 10.2
Sample font file

Typeface	*Roman*	*Bold*	*Italic*	*Bold Italic*
Video Gael				
SR-I	19	29	-	-
SR-II	18	28	22	-
Video Gothic				
SR-I	11	3	-	-
SR-II	10	2	-	-
SR-III	66	68	-	-
SR-IV	-	67	-	-
Video Greek				
SR-I	99	-	-	-
SR-II	98	-	-	-
Video News Gothic				
SR-I	65	63	-	-
SR-II	64	62	-	-
Video Technica				
SR-I	13	9	-	-
SR-II	12	8	-	-
Video Times Roman				
SR-I	5,25	7	15	-
SR-II	4,24	6	14	26
SR-III	166	-	-	-
Video Display Math				
SR-IV	104	-	-	-

All fonts which appear in this table are displayed in Appendix C.

Manuscript markup

The total job comprises a number of *logical units.* Each chapter is generally considered to be one logical unit; front and back matter may be assigned as additional logical units preceding or following the body of the book as the individual user desires.

The job specification will, in general, be the first part of take one of the first logical unit. If chapter one is the first logical unit, the definition of **Z2** as the title of chapter one and the assignment of a value one less than the number of the opening page to **pn** may be included in the job specification. Since **C1** increments the page number counter, **pn** must be assigned a value one less than the number desired for the first page.

Chapter four of *Strategy for the Americas* is used for sample pages. The marked-up manuscript is shown in Figure 10.4. The character $\overset{\text{T}}{\text{H}}$ represents a thin space and is the character which indentifies the keyboard, and the associated translation table in the processor,

Figure 10.3
Job specification for *Strategy for the Americas*

Allowance for line numbers		[gv122,12;	(*)
Column measure		gv59,288;	24 picas
Nominal lines per page		gv60,38;	38 lines per page
Paragraph indention		gv76,1;	1 em
Typeface		gv41,18;	Caledonia†
Body of the book		gv11,10;gv26,12;	10/12
Secondary text—extracts	font/size	gv12,9;gv27,11;gv42,18;	9/11 Caledonia†
	space above/below	gv102,4;gv104,4;	(* 4 pts. above and below)
	indention	gv61,2;	2 ems
Footnotes	font/size	gv14,8;gv29,10;gv44,18;	8/10 Caledonia†
	font/size for citation	gv25,5;gv55,19;	(* 5 pt. Caledonia‡)
	text-footnote minimum space	gv172,6;	(* 6 pts.)
Folio	font/size	gv15,10;gv45,18;	10 pt. Caledonia†
	space above drop folio	gv94,2;	2 pts. above
First chapter head (number)	font/size	gv17,24;gv32,24;gv47,166;	24 pt. Caslon Bold§
	space above	gv107,30;	2½ picas above
	indention	gv71,7;	7 pica indention
Second chapter head (title)	font/size	gv18,18;gv33,36;gv48,166;	18 pt. Caslon Bold§
	space above	gv108,24;	2 picas above, 1½ picas between lines
	indention	gv73,7;	7 pica indention
Chapter opening	sink below title	gv110,24;	2 picas below title
	opening line indention	gv56,7;	7 pica indention
First side head	font/size	gv20,10;gv35,12;gv50,28;	10/12 Caledonia Bold¶
	space above/below	gv111,12;gv113,6;	12 pts.above, 6 pts. below
	indention	gv77,0;	no
Ssecond side head	font/size	gv21,10;gv36,12;gv51,22;	10/12 Caledonia Italic**
	space above/below	gv114,10;gv116,2;	10 pts. above, 2 pts below
	indention	gv78,0;	no
Paragraph follows side heads		gv80,1;	yes
Running heads	left-hand text	sy,Z1[STRATEGY FOR THE AMERICAS];	Book title
	right-hand text		Chapter title††
	font/size	gv16,10;gv46,18;	10 pt Caledonia small caps†
	space below	gv95,14]	14 pts. below
Text block *when bottom* procedure		[sy,Y1[[C2]];	(*)
Running head		sy,Y2[[G1]];	(*)
Footnote block *when bottom* procedure		sy,Y3[[C3]];	(*)
Drop folio, chapter opening		sy,Y4[[G6]]]	(*)
Synonyms for true italic	to italic	[sy,Z8[[tf,22]];	(*)
	to roman	sy,Z9[[tf,18]]]	(*)
Style generator		[I8]	(*)

---

* These items are not a part of the conventional style specification.

† Use Video Gael, SR-II.

‡ Use Video Gael, SR-I.

§ Use Video Times Roman, SR-III.

¶ Use Video Gael Bold, SR-II.

** Use Video Gael Italic, SR-II.

†† Define synonym at each chapter opening, prior to call of [c1]

used in generating these samples. The job number is 301 and the logical unit number and take number are both one. The data tape for this sample includes the job specification of Figure 10.3 between the first and second lines of codes at the top of manuscript page 76 in Figure 10.4. Normally, only the definition of **Z2** will appear at the beginning of each new chapter prior to the call of the pagination algorithm.

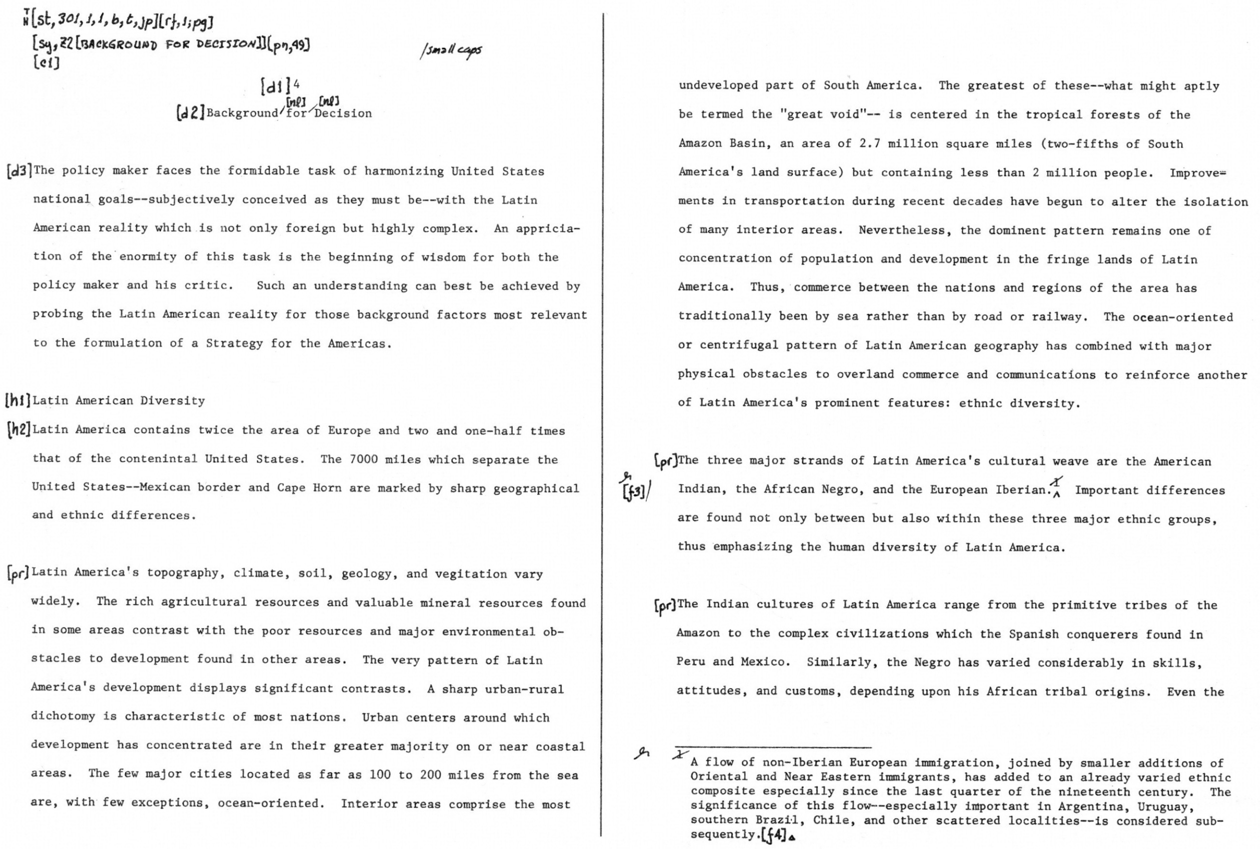
Following **C1**, the format labels for chapter number, chapter title, and chapter opening are written on the manuscripts as shown in the figure; the chapter title is specified to be set capital and lower case broken into "lines of nearly equal length" by the mark-up person.

Mark-up for the remainder comprises primarily insertion of paragraph breaks. Side heads are introduced by **H1** and primary text is resumed following the head by **H2** (**H3** and **H4** for second level side heads). The general convention for the relationship of text, control, and wordspace is that controls precede text words with no intervening wordspace. The wordspace precedes control. There can be no absolute rules and in Page-1 markup one must be aware of exceptions to this general rule.

Citation of footnotes by the format label **F3** and return to primary text by the format label **F4** is a case in point. The footnote citation is typeset by the *typeset decimal number* code in conjunction with the footnote citation counter. In order that spacing be correct in the typeset page, there must not be a wordspace between the text word to which the footnote citation is attached and the format label **F3**; the wordspace required before the next text word must follow the format label **F4**.

In the center paragraph of manuscript page 77 (Fig-

Figure 10.4
Manuscript for *Strategy for the Americas*



T
H[st,301,1,1,b,t,jp][rj,1;pg]
[sy,Z2[BACKGROUND FOR DECISION]](pn,49)
[c1]

/small caps

[d1]⁴

[d2]Background/[nl]for/[nl]Decision

[d3]The policy maker faces the formidable task of harmonizing United States national goals--subjectively conceived as they must be--with the Latin American reality which is not only foreign but highly complex. An appriciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

[h1]Latin American Diversity

[h2]Latin America contains twice the area of Europe and two and one-half times that of the conteninal United States. The 7000 miles which separate the United States--Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

[pr]Latin America's topography, climate, soil, geology, and vegitation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's development displays significant contrasts. A sharp urban-rural dichotomy is characteristic of most nations. Urban centers around which development has concentrated are in their greater majority on or near coastal areas. The few major cities located as far as 100 to 200 miles from the sea are, with few exceptions, ocean-oriented. Interior areas comprise the most undeveloped part of South America. The greatest of these--what might aptly be termed the "great void"-- is centered in the tropical forests of the Amazon Basin, an area of 2.7 million square miles (two-fifths of South America's land surface) but containing less than 2 million people. Improve= ments in transportation during recent decades have begun to alter the isolation of many interior areas. Nevertheless, the dominent pattern remains one of concentration of population and development in the fringe lands of Latin America. Thus, commerce between the nations and regions of the area has traditionally been by sea rather than by road or railway. The ocean-oriented or centrifugal pattern of Latin American geography has combined with major physical obstacles to overland commerce and communications to reinforce another of Latin America's prominent features: ethnic diversity.

[pr]The three major strands of Latin America's cultural weave are the American Indian, the African Negro, and the European Iberian.~~1~~[f3]/ Important differences are found not only between but also within these three major ethnic groups, thus emphasizing the human diversity of Latin America.

[pr]The Indian cultures of Latin America range from the primitive tribes of the Amazon to the complex civilizations which the Spanish conquerers found in Peru and Mexico. Similarly, the Negro has varied considerably in skills, attitudes, and customs, depending upon his African tribal origins. Even the

---

~~1~~A flow of non-Iberian European immigration, joined by smaller additions of Oriental and Near Eastern immigrants, has added to an already varied ethnic composite especially since the last quarter of the nineteenth century. The significance of this flow--especially important in Argentina, Uruguay, southern Brazil, Chile, and other scattered localities--is considered subsequently.[f4]▲

ure 10.4), the footnote citation is deleted and **F3** follows the word "Iberian." without a wordspace. The footnote text follows immediately and is in turn followed by **F4** to close the footnote and resume primary text. The wordspace necessary between the typeset citation in text and the word "Important" must follow **F4**. The space is represented by the Δ following **[F4]** at the bottom of manuscript page 77 in Figure 10.4. The typeset pages are shown in Fgure 10.5.

A technical book

MAINTAINABILITY: A Major Element of System Effectiveness was published by John Wiley and Sons in 1966. The style specification is shown in Figure 10.6. This title will be used to illustrate a mixture of job oriented synonyms and global format statements to handle the typesetting of a book. The essential differences between "Maintainability" and "Strategy" are the requirement to allocate white space for graphics, the use of special characters to cite footnotes, specialized chapter heading requirements not covered by the format manual, and the presence of display mathematics in Maintainability.

The white space and footnote citation requirements are quite easily disposed of by selecting **C6** as the pagination algorithm and **F6** and **F7** for footnotes; **C9** becomes the citation for white space. The first level sub-head is a center head which will be satisfied by **P1** and **P2**. The second and third level sub-heads are side heads flush left) for which **H3** through **H6** will be used. The fourth level sub-head is a run-in side head which will be handled by **H7** and **H8**.

Chapter heading

The sketch shown in Figure 10.7 illustrates the chapter heading desired; the synonyms developed in the figure will generate it. The style specification for the book does not give detailed instructions for the

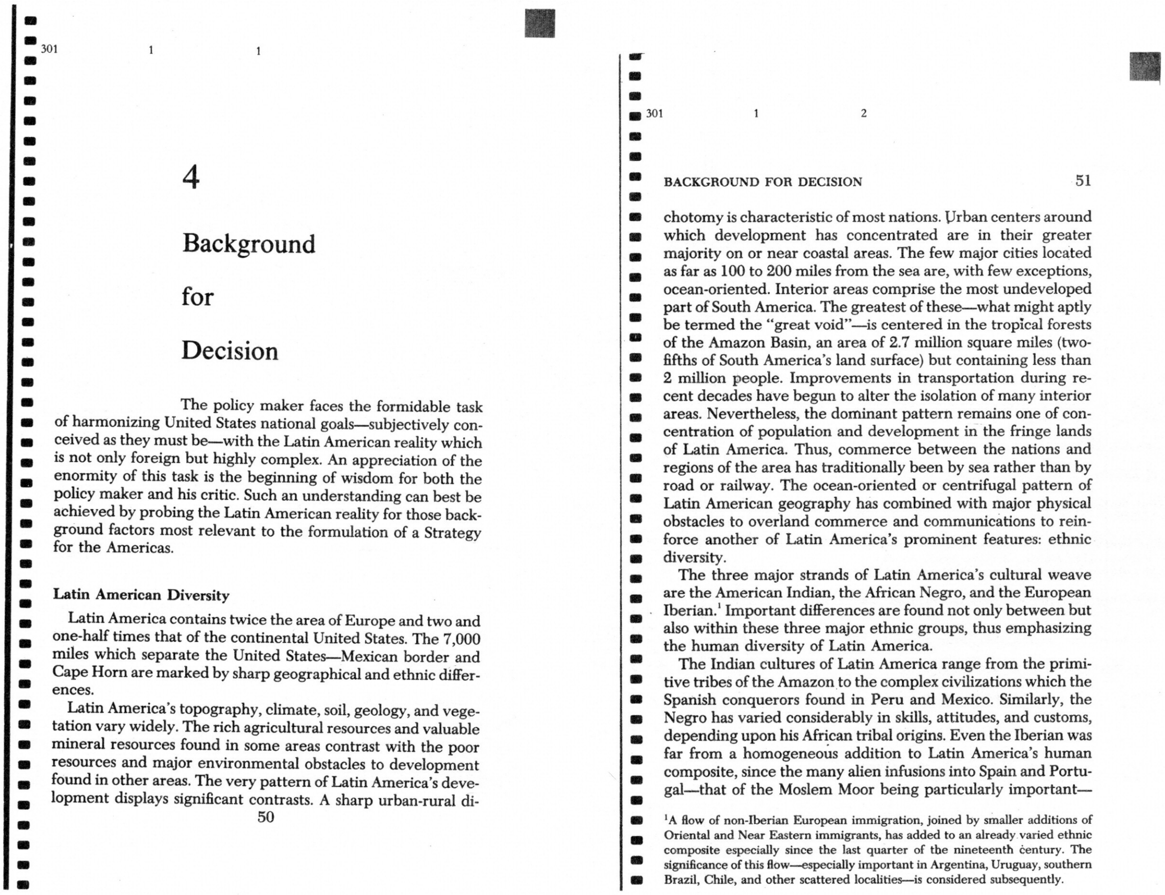


Figure 10.5
Sample pages from *Strategy for the Americas*

301 1 1

4

Background

for

Decision

The policy maker faces the formidable task of harmonizing United States national goals—subjectively conceived as they must be—with the Latin American reality which is not only foreign but highly complex. An appreciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

Latin American Diversity

Latin America contains twice the area of Europe and two and one-half times that of the continental United States. The 7,000 miles which separate the United States—Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

Latin America's topography, climate, soil, geology, and vegetation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's development displays significant contrasts. A sharp urban-rural di-

50

301 1 2

BACKGROUND FOR DECISION 51

chotomy is characteristic of most nations. Urban centers around which development has concentrated are in their greater majority on or near coastal areas. The few major cities located as far as 100 to 200 miles from the sea are, with few exceptions, ocean-oriented. Interior areas comprise the most undeveloped part of South America. The greatest of these—what might aptly be termed the "great void"—is centered in the tropical forests of the Amazon Basin, an area of 2.7 million square miles (two-fifths of South America's land surface) but containing less than 2 million people. Improvements in transportation during recent decades have begun to alter the isolation of many interior areas. Nevertheless, the dominant pattern remains one of concentration of population and development in the fringe lands of Latin America. Thus, commerce between the nations and regions of the area has traditionally been by sea rather than by road or railway. The ocean-oriented or centrifugal pattern of Latin American geography has combined with major physical obstacles to overland commerce and communications to reinforce another of Latin America's prominent features: ethnic diversity.

The three major strands of Latin America's cultural weave are the American Indian, the African Negro, and the European Iberian.[1] Important differences are found not only between but also within these three major ethnic groups, thus emphasizing the human diversity of Latin America.

The Indian cultures of Latin America range from the primitive tribes of the Amazon to the complex civilizations which the Spanish conquerors found in Peru and Mexico. Similarly, the Negro has varied considerably in skills, attitudes, and customs, depending upon his African tribal origins. Even the Iberian was far from a homogeneous addition to Latin America's human composite, since the many alien infusions into Spain and Portugal—that of the Moslem Moor being particularly important—

[1] A flow of non-Iberian European immigration, joined by smaller additions of Oriental and Near Eastern immigrants, has added to an already varied ethnic composite especially since the last quarter of the nineteenth century. The significance of this flow—especially important in Argentina, Uruguay, southern Brazil, Chile, and other scattered localities—is considered subsequently.

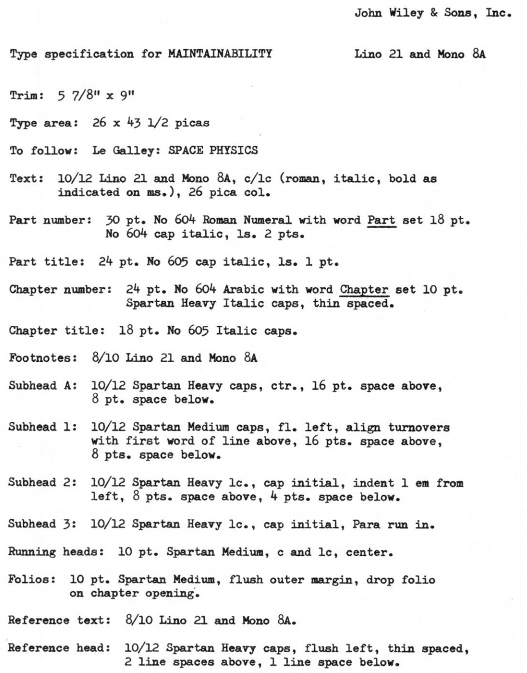
chapter opening page; rather it specifies the typeface and sizes and refers to an existing book (Le Galley: *SPACE PHYSICS*) a copy of which was given to the typesetter along with the style specification.

In general, the typesetter is allowed considerable latitude in making-up the chapter opening page, as he is with space around sub-heads and space between text and footnotes. This latitude allows the maker-up to avoid potentially awkward situations such as widows, or even carryover footnotes, on succeeding chapter pages. Fairly wide variation in the spacing on the chapter opening pages of "MAINTAINABILITY" is evident and there is of course, nothing wrong with this. In Page–1, however, and using generalized procedures, all of the chapter openings will be alike unless one or more are altered in a subsequent correction run.

The sketch shown in Figure 10.7 is taken from Chapter 2 of the published book. The sketch shows that the word "CHAPTER", set in ten point, and the chapter number, set in 24 point, are "top aligned" with eight points above. The position for the word chapter is specified by the combination of a *non-justifiable down* command and a *baseline up* command. The "down thirty-two" drops the baseline of the first line to be typeset such that the top of a 24 point character will be eight points below the top of the nominal type page. The "up 488" raises the baseline fourteen points and the top of the ten point character will also be eight points below the top of the type page.

These two commands are followed by codes to establish ten point type, an 18 point body lead, and typeface eight which is Video Technica Bold and the *oblique* mode are used here in place of Spartan Heavy Italic. Since down commands are used throughout the code strings being developed (except for the *new line* code used to break the chapter title), body lead is not used by the Page-1 processor. It is used to provide proper line spacing between chapter title lines, the

Figure 10.6
Style specification for *MAINTAINABILITY: A Major Element of System Effectiveness*



John Wiley & Sons, Inc.

Type specification for MAINTAINABILITY   Lino 21 and Mono 8A

Trim: 5 7/8" x 9"

Type area: 26 x 43 1/2 picas

To follow: Le Galley: SPACE PHYSICS

Text: 10/12 Lino 21 and Mono 8A, c/lc (roman, italic, bold as indicated on ms.), 26 pica col.

Part number: 30 pt. No 604 Roman Numeral with word Part set 18 pt. No 604 cap italic, ls. 2 pts.

Part title: 24 pt. No 605 cap italic, ls. 1 pt.

Chapter number: 24 pt. No 604 Arabic with word Chapter set 10 pt. Spartan Heavy Italic caps, thin spaced.

Chapter title: 18 pt. No 605 Italic caps.

Footnotes: 8/10 Lino 21 and Mono 8A

Subhead A: 10/12 Spartan Heavy caps, ctr., 16 pt. space above, 8 pt. space below.

Subhead 1: 10/12 Spartan Medium caps, fl. left, align turnovers with first word of line above, 16 pts. space above, 8 pts. space below.

Subhead 2: 10/12 Spartan Heavy lc., cap initial, indent 1 em from left, 8 pts. space above, 4 pts. space below.

Subhead 3: 10/12 Spartan Heavy lc., cap initial, Para run in.

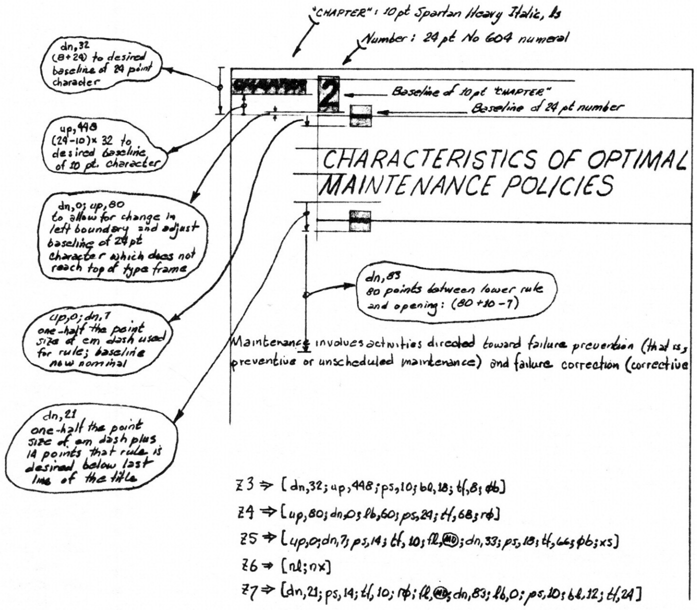
Running heads: 10 pt. Spartan Medium, c and lc, center.

Folios: 10 pt. Spartan Medium, flush outer margin, drop folio on chapter opening.

Reference text: 8/10 Lino 21 and Mono 8A.

Reference head: 10/12 Spartan Heavy caps, flush left, thin spaced, 2 line spaces above, 1 line space below.

Figure 10.7
Chapter heading for *MAINTAINABILITY: A Major Element of System Effectiveness*


"CHAPTER": 10 pt Spartan Heavy Italic, ls
Number: 24 pt No 604 numeral
dn,32 (8+24) to desired baseline of 24 point character
Baseline of 10 pt "CHAPTER"
Baseline of 24 pt number
up,448 (24-10)× 32 to desired baseline of 10 pt. character
CHARACTERISTICS OF OPTIMAL MAINTENANCE POLICIES
dn,0; up,80 to allow for change in left boundary and adjust baseline of 24 pt character which does not reach top of type frame
dn,83 80 points between lower rule and opening: (80 +10 −7)
up,0; dn,7 one-half the point size of em dash used for rule; baseline now nominal
Maintenance involves activities directed toward failure prevention (that is, preventive or unscheduled maintenance) and failure correction (corrective
dn,21 one-half the point size of em dash plus 14 points that rule is desired below last line of the title

```
Z3 ⇒ [dn,32;up,448;ps,10;bl,18;tf,8;ob]
Z4 ⇒ [up,80;dn,0;lb,60;ps,24;tf,68;rφ]
Z5 ⇒ [up,0;dn,7;ps,14;tf,10;fl,MD;dn,33;ps,18;tf,66;φb;xs]
Z6 ⇒ [nl;nx]
Z7 ⇒ [dn,21;ps,14;tf,10;rφ;fl,MD;dn,83;lb,0;ps,10;bl,12;tf,24]
```

value required there is arbitrarily specified here and left in force until font and size are specified for body text. This string of codes is given the synonym name **z3**; it is shown in Figure 10.7.

After execution of **z3**, the word "chapter" is typeset. Next, the baseline is adjusted to account for the fact that the number 2, in the font used, does not extend to the top of the type frame and the line is terminated with a "down zero" command. This provides the opportunity to move the left boundary to properly position the chapter number and establish the line length for the title and the horizontal rules which bound it. Left boundary sixty was chosen as a compromise between the space desired between the word chapter and the number and the weight of the horizontal rule which is determined by point size since the rule is generated using the *fill-to-measure* with em dashes. The chapter title is specified to be set in eighteen point No. 605 italic caps. (Video Gothic is used in this sample.) The same typeface (different size range) will be used to generate the horizontal rules. A left boundary of sixty yields a measure of 252 points for the chapter title and point size fourteen is an even numbered sub-multiple of 252 points. The code string, given the synonym name **z4**, adjusts the baseline and then changes the left boundary to sixty, the point size to 24 and the typeface to 68 for the chapter number, and restores the roman/oblique mode to roman.

The horizontal rule (in the published book) above the chapter title seemed to be coincident with the lower edge of the 24 point type frame of the chapter number. This relationship is specified at the beginning of **z5** by a "down seven" command and the change in point size to fourteen. Typeface is then changed to ten (Video Gothic, Size Range II, which includes point size fourteen) and the horizontal rule is generated with the *fill-to-measure* command. Twenty-four points between the upper horizontal rule and the first line of the chapter title are provided by "down thirty-three". Next, the point size is changed to 18 and the typeface to 66 for the chapter title. Finally, the roman/oblique mode is set to *oblique* and the *expand-to-fill-measure* mode is specified in order that the first line of the title (actually all lines except the last) will be expanded.

Chapter title lines will be broken by *new line* codes and all lines will be expanded to the 252 point measure bounded by the horizontal rules. The synonym **z6** precedes the last line of the title and cancels the *expand-to-fill-measure* in addition to providing the line break.

The lower horizontal rule is to be fourteen points below the last line of the title. A "down twenty-one" accomplishes this. Then the point size and type face are changed to fourteen and ten respectively, for the horizontal rule and the horizontal rule is again generated with the *fill-to-measure* code. An 80 point sink from the lower horizontal rule to the first line of body text is provided by "down eighty-three". The synonym **z7** is concluded with codes to restore the left boundary to zero and to change the font and size to 10/12 Video Times Roman. Typeface 24 is used in this job because it has several sub-sets of selected math characters.

The five synonyms **z3** through **z7** will set the chapter openings according to the rationale just developed. They will be a part of the job specification and their use in manuscript markup will be shown subsequently. It is perhaps interesting to note that **z3** through **z5** could be combined to form one string which would handle everything prior to entry of the chapter title. The word "chapter", letterspaced (a cumbersome thing in Page–1 since it must be done with fixed horizontal spaces) can be explicitly stated and the chapter number can be included implicitly by use of the *set decimal number* code. A general variable, for example **gv9** could be initialized to one in the job specification. The codes **...;dc,gv9;ad,gv9,gv9,1;...** then will cause the processor to typeset the chapter number and then increment **gv9** for the next call of the resulting synonym.

Mathematics

The typesetting of display mathematics is not covered in this treatment of Page–1; yet the language is demonstrably flexible and general enough that procedures can be written to typeset display mathematics. The obvious difficulty is that a very large number of format statements are required to handle a significant sub-set of all math. The problem of character sets required to support mathematical typesetting further complicates matters. The typesetting of display mathematics is no mean undertaking under any conditions however. It seems likely that a set of procedures required by a given mathematical or scientific discipline might be small enough to be manageable. To this end "MAINTAINABILITY" was chosen as a sample because its display math requirements are significant—neither overwhelming nor trivial.

Though the synonyms developed to handle the requirements of this manuscript are not presented here, a brief sample is included in the sample pages of "MAINTAINABILITY", shown subsequently, to demonstrate the power and flexibility of the language.

Figure 10.8
Job specification for *MAINTAINABILITY: A Major Element of System Effectiveness*

Allowance for line numbers		[gv122,12;	(*)
Column measure		gv59,312;	26 picas
Nominal lines per page		gv60,42;	42 lines per page
Paragraph indention		gv76,1;	1 em
Typeface		gv41,24;	Lino 21 and Mono 8A†
Body of the book		gv11,10;gv26,12;	10/12
secondary text	font/size	gv12,10;gv27,12;gv42,24;	10/12
	indention	gv61,0;	None
	space above/below	gv102,10;gv104,10;	(* 10 pts. above/below)
Footnotes	font/size	gv14,8;gv29,10;gv44,24;	8/10
	text-footnote minumum space	gv172,6;	(* 6 pts.)
Folio	font/size	gv15,10;gv45,18;	10 pt. Spartan Medium‡
	space above drop folio	gv94,4;	4 pts. above
Subhead A	font/size	gv20,10;gv35,12;gv50,8;	10/12 Spartan Heavy§
	space above/below	gv111,16;gv113,8;	16 pts.above, 8 pts. below
	indention	gv77,1;	yes
Subhead 1	font/size	gv21,10;gv36,12;gv51,12;	10/12 Spartan Medium‡
	space above/below	gv114,16;gv116,8;	16 pts. above, 8 pts. below
	indention	gv78,1;	yes
Subhead 2	font/size	gv22,10;gv37,12;gv52,8;	10/12 Spartan Heavy§
	space above/below	gv117,8;gv119,4;	8 pts. above, 4 pts. below
	indention	gv79,1;	yes
Subhead 3	font/size	gv13,10;gv28,12;gv43,8;	10/12 Spartan Heavy§
	space above	gv106,0;	none, paragraph run in
	space between head and text	gv81,1;	yes
Paragraph follows side heads		gv80,1;	yes
Running heads	left-hand text		Part title¶
	right-hand text		Chapter title¶
	font/size	gv16,10;gv46,18;	10 pt Spartan Medium‡
	space below	gv95,10]	10 pts. below
Text block *when bottom* procedure		[sy,Y1[[C7]];	(*)
Running head		sy,Y2[[G5]];	(*)
Footnote block *when bottom* procedure		sy,Y3[[C8]];	(*)
Drop folio, chapter opening		sy,Y4[[G6]]]	(*)
Synonyms for chapter heading			
	the word chapter	[sy,Z3[[dn,32;up,448;ps,10;bl,18;tf,8;ob]];	(*, **)
	number	sy,Z4[[up,80;dn,0;lb,60;ps,24;tf,68;ro]];	(*, **)
	title	sy,Z5[[up,0;dn,7;ps,14;tf,10;fl,—;dn,33;ps,18;tf,66;ob;xs]];	(*, **)
	between chapter lines	sy,Z6[[nl;nx]];	(*, **)
	opening	sy,Z7[[dn,21;ps,14;tf,10;ro;fl,—;dn,83;lb,0;ps,10;bl,12;tf,24]]]	(*, **)
Synonyms for true italic	to italic	[sy,Z8[[tf,14]];	(*)
	to roman	sy,Z9[[tf,24]]]	(*)
Style generator		[I8]	(*)

---

* These items are not a part of the conventional style specification.

† Use Video Times Roman, SR-II.

‡ Use Video Technica Medium, SR-II.

§ Use Video Technica Bold, SR-II.

¶ Define synonyms at each part and chapter opening.

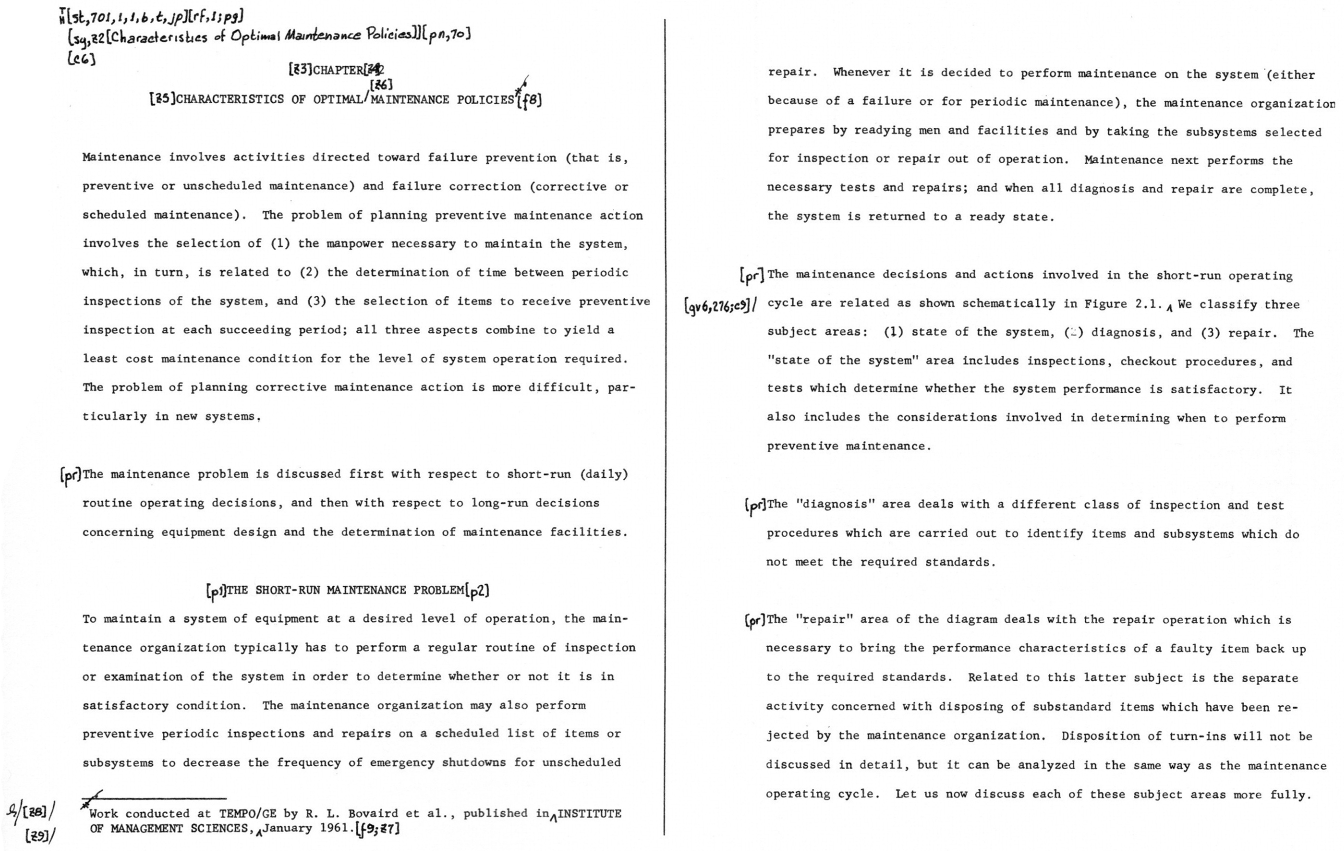
** See Figure 10.8 and text.

The job specification

The job specification for "MAINTAINABILITY" is shown in Figure 10.8; an item by item comparison of Figures 10.6 and 10.8, with reference to the format manual as may be required, is detailed below:

- The job specification begins with an allowance of twelve points for proof numbers;
- Column measure is 26 picas—312 points;
- A nominal text page comprises 42 lines;
- The standard paragraph indention is one em;
- The typeface for the body of the book is Lino 21 and Mono 8A—Video Times Roman has been substituted;
- The body of the book is set 10/12;
- Secondary text is set 10/12 (Video Times Roman, Size Range II), full measure with ten points above and below (in this job **J1** and **J2** are used for lists of items set out in text);
- Footnotes are set 8/10 (Video Times Roman, Size Range II)—They are separated from text by six points;
- The folio is ten point Spartan Medium (Video Technica Medium, Size Range II has been substituted) and four points are specified above the chapter opening drop-folio;
- Sub-head A is specified to be 10/12 Spartan Heavy (Video Technica Bold Size Range II has been substituted) with sixteen points above and eight points below—Sub-head A is a center head and no indention is desired;
- Sub-head 1 is specified to be 10/12 Spartan Medium (Video Technica Medium) with sixteen points above and eight points below—the side head is given the standard paragraph indention;
- Sub-head 2 is specified to be 10/12 Spartan Bold (Video Technica Bold) with eight points above and four below—the side head is given the standard paragraph indention;

Figure 10.9
Manuscript for *MAINTAINABILITY: A Major Element of System Effectiveness*



T/H[st,701,1,1,b,t,jp][rf,1;pg]
[sy,z2[Characteristics of Optimal Maintenance Policies]][pn,70]
[c6]

[z3]CHAPTER[z4]2

[z5]CHARACTERISTICS OF OPTIMAL/[z6]MAINTENANCE POLICIES*[f8]

Maintenance involves activities directed toward failure prevention (that is, preventive or unscheduled maintenance) and failure correction (corrective or scheduled maintenance). The problem of planning preventive maintenance action involves the selection of (1) the manpower necessary to maintain the system, which, in turn, is related to (2) the determination of time between periodic inspections of the system, and (3) the selection of items to receive preventive inspection at each succeeding period; all three aspects combine to yield a least cost maintenance condition for the level of system operation required. The problem of planning corrective maintenance action is more difficult, particularly in new systems.

[pr]The maintenance problem is discussed first with respect to short-run (daily) routine operating decisions, and then with respect to long-run decisions concerning equipment design and the determination of maintenance facilities.

[p1]THE SHORT-RUN MAINTENANCE PROBLEM[p2]

To maintain a system of equipment at a desired level of operation, the maintenance organization typically has to perform a regular routine of inspection or examination of the system in order to determine whether or not it is in satisfactory condition. The maintenance organization may also perform preventive periodic inspections and repairs on a scheduled list of items or subsystems to decrease the frequency of emergency shutdowns for unscheduled repair. Whenever it is decided to perform maintenance on the system (either because of a failure or for periodic maintenance), the maintenance organization prepares by readying men and facilities and by taking the subsystems selected for inspection or repair out of operation. Maintenance next performs the necessary tests and repairs; and when all diagnosis and repair are complete, the system is returned to a ready state.

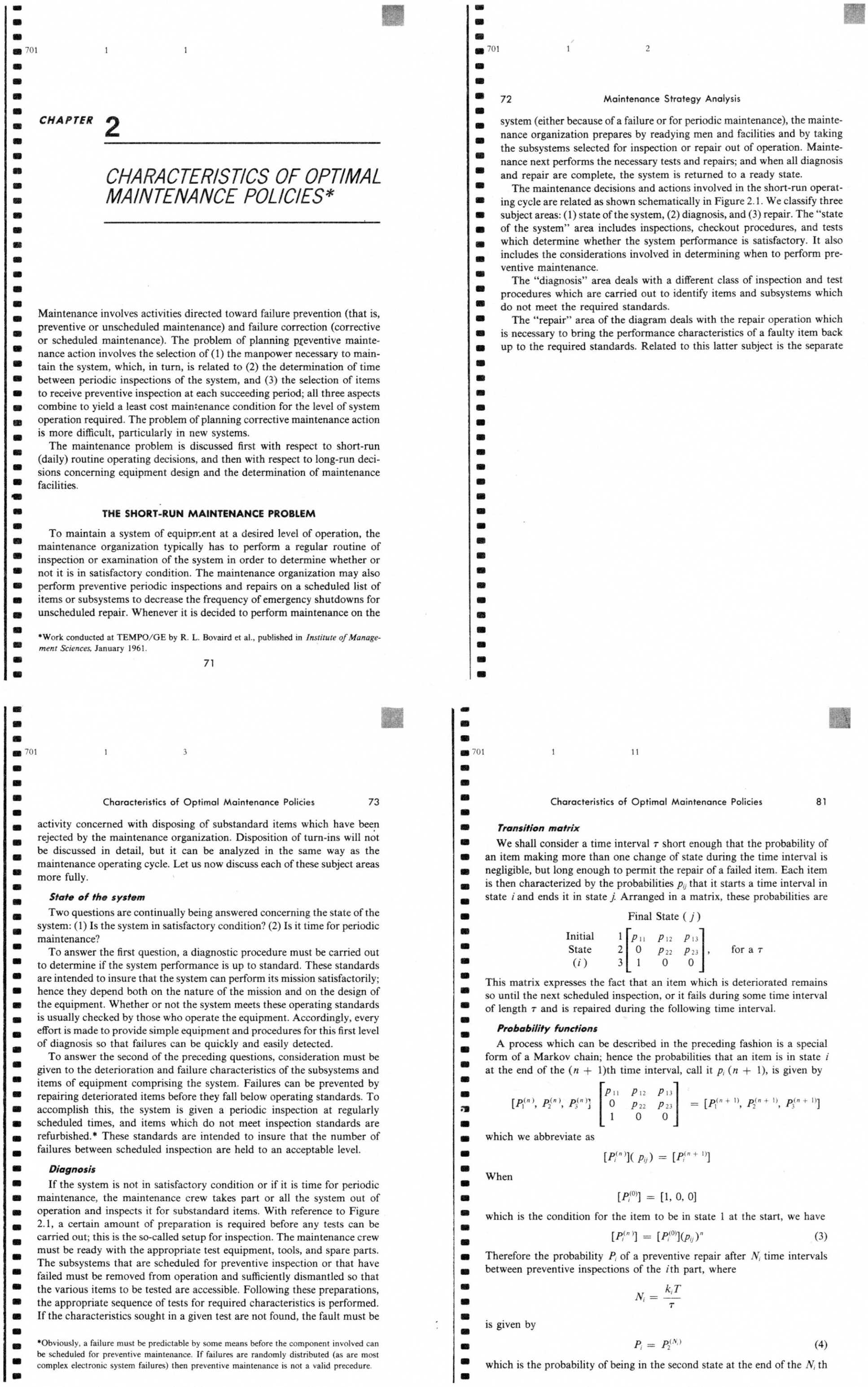
[pr]The maintenance decisions and actions involved in the short-run operating cycle are related as shown schematically in Figure 2.1. [gv6,276;c9]/ We classify three subject areas: (1) state of the system, (2) diagnosis, and (3) repair. The "state of the system" area includes inspections, checkout procedures, and tests which determine whether the system performance is satisfactory. It also includes the considerations involved in determining when to perform preventive maintenance.

[pr]The "diagnosis" area deals with a different class of inspection and test procedures which are carried out to identify items and subsystems which do not meet the required standards.

[pr]The "repair" area of the diagram deals with the repair operation which is necessary to bring the performance characteristics of a faulty item back up to the required standards. Related to this latter subject is the separate activity concerned with disposing of substandard items which have been rejected by the maintenance organization. Disposition of turn-ins will not be discussed in detail, but it can be analyzed in the same way as the maintenance operating cycle. Let us now discuss each of these subject areas more fully.

ℓ/[z8]/ *Work conducted at TEMPO/GE by R. L. Bovaird et al., published in INSTITUTE
[z9]/ OF MANAGEMENT SCIENCES, January 1961.[f9;z7]

Figure 10.10
Sample pages from *MAINTAINABILITY: A Major Element of System Effectiveness*



701 1 1

CHAPTER 2

*CHARACTERISTICS OF OPTIMAL MAINTENANCE POLICIES**

Maintenance involves activities directed toward failure prevention (that is, preventive or unscheduled maintenance) and failure correction (corrective or scheduled maintenance). The problem of planning preventive maintenance action involves the selection of (1) the manpower necessary to maintain the system, which, in turn, is related to (2) the determination of time between periodic inspections of the system, and (3) the selection of items to receive preventive inspection at each succeeding period; all three aspects combine to yield a least cost maintenance condition for the level of system operation required. The problem of planning corrective maintenance action is more difficult, particularly in new systems.

The maintenance problem is discussed first with respect to short-run (daily) routine operating decisions, and then with respect to long-run decisions concerning equipment design and the determination of maintenance facilities.

THE SHORT-RUN MAINTENANCE PROBLEM

To maintain a system of equipment at a desired level of operation, the maintenance organization typically has to perform a regular routine of inspection or examination of the system in order to determine whether or not it is in satisfactory condition. The maintenance organization may also perform preventive periodic inspections and repairs on a scheduled list of items or subsystems to decrease the frequency of emergency shutdowns for unscheduled repair. Whenever it is decided to perform maintenance on the

---

*Work conducted at TEMPO/GE by R. L. Bovaird et al., published in *Institute of Management Sciences*, January 1961.

71

701 1 2

72 Maintenance Strategy Analysis

system (either because of a failure or for periodic maintenance), the maintenance organization prepares by readying men and facilities and by taking the subsystems selected for inspection or repair out of operation. Maintenance next performs the necessary tests and repairs; and when all diagnosis and repair are complete, the system is returned to a ready state.

The maintenance decisions and actions involved in the short-run operating cycle are related as shown schematically in Figure 2.1. We classify three subject areas: (1) state of the system, (2) diagnosis, and (3) repair. The "state of the system" area includes inspections, checkout procedures, and tests which determine whether the system performance is satisfactory. It also includes the considerations involved in determining when to perform preventive maintenance.

The "diagnosis" area deals with a different class of inspection and test procedures which are carried out to identify items and subsystems which do not meet the required standards.

The "repair" area of the diagram deals with the repair operation which is necessary to bring the performance characteristics of a faulty item back up to the required standards. Related to this latter subject is the separate

701 1 3

Characteristics of Optimal Maintenance Policies 73

activity concerned with disposing of substandard items which have been rejected by the maintenance organization. Disposition of turn-ins will not be discussed in detail, but it can be analyzed in the same way as the maintenance operating cycle. Let us now discuss each of these subject areas more fully.

State of the system

Two questions are continually being answered concerning the state of the system: (1) Is the system in satisfactory condition? (2) Is it time for periodic maintenance?

To answer the first question, a diagnostic procedure must be carried out to determine if the system performance is up to standard. These standards are intended to insure that the system can perform its mission satisfactorily; hence they depend both on the nature of the mission and on the design of the equipment. Whether or not the system meets these operating standards is usually checked by those who operate the equipment. Accordingly, every effort is made to provide simple equipment and procedures for this first level of diagnosis so that failures can be quickly and easily detected.

To answer the second of the preceding questions, consideration must be given to the deterioration and failure characteristics of the subsystems and items of equipment comprising the system. Failures can be prevented by repairing deteriorated items before they fall below operating standards. To accomplish this, the system is given a periodic inspection at regularly scheduled times, and items which do not meet inspection standards are refurbished.* These standards are intended to insure that the number of failures between scheduled inspection are held to an acceptable level.

Diagnosis

If the system is not in satisfactory condition or if it is time for periodic maintenance, the maintenance crew takes part or all the system out of operation and inspects it for substandard items. With reference to Figure 2.1, a certain amount of preparation is required before any tests can be carried out; this is the so-called setup for inspection. The maintenance crew must be ready with the appropriate test equipment, tools, and spare parts. The subsystems that are scheduled for preventive inspection or that have failed must be removed from operation and sufficiently dismantled so that the various items to be tested are accessible. Following these preparations, the appropriate sequence of tests for required characteristics is performed. If the characteristics sought in a given test are not found, the fault must be

---

*Obviously, a failure must be predictable by some means before the component involved can be scheduled for preventive maintenance. If failures are randomly distributed (as are most complex electronic system failures) then preventive maintenance is not a valid precedure.

701 1 11

Characteristics of Optimal Maintenance Policies 81

Transition matrix

We shall consider a time interval τ short enough that the probability of an item making more than one change of state during the time interval is negligible, but long enough to permit the repair of a failed item. Each item is then characterized by the probabilities p_{ij} that it starts a time interval in state i and ends it in state j. Arranged in a matrix, these probabilities are

$$\begin{array}{cc} & \text{Final State } (j) \\ \begin{array}{c}\text{Initial}\\ \text{State}\\ (i)\end{array} \begin{array}{c}1\\2\\3\end{array} & \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ 0 & p_{22} & p_{23} \\ 1 & 0 & 0 \end{bmatrix}, \quad \text{for a } \tau \end{array}$$

This matrix expresses the fact that an item which is deteriorated remains so until the next scheduled inspection, or it fails during some time interval of length τ and is repaired during the following time interval.

Probability functions

A process which can be described in the preceding fashion is a special form of a Markov chain; hence the probabilities that an item is in state i at the end of the $(n + 1)$th time interval, call it $p_i\,(n + 1)$, is given by

$$[P_1^{(n)}, P_2^{(n)}, P_3^{(n)}] \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ 0 & p_{22} & p_{23} \\ 1 & 0 & 0 \end{bmatrix} = [P_1^{(n+1)}, P_2^{(n+1)}, P_3^{(n+1)}]$$

which we abbreviate as

$$[P_i^{(n)}](p_{ij}) = [P_i^{(n+1)}]$$

When

$$[P_i^{(0)}] = [1, 0, 0]$$

which is the condition for the item to be in state 1 at the start, we have

$$[P_i^{(n)}] = [P_i^{(0)}](p_{ij})^n \qquad (3)$$

Therefore the probability P_i of a preventive repair after N_i time intervals between preventive inspections of the ith part, where

$$N_i = \frac{k_i T}{\tau}$$

is given by

$$P_i = P_2^{(N_i)} \qquad (4)$$

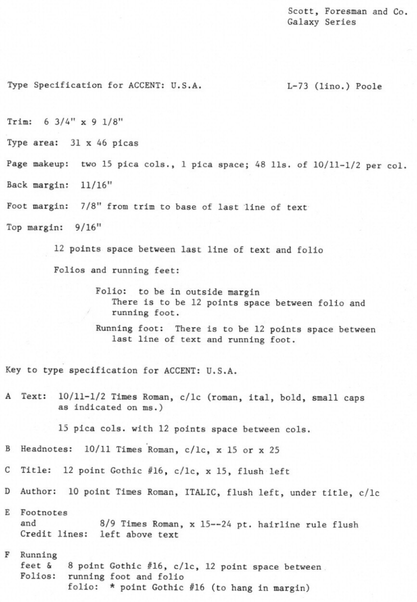
which is the probability of being in the second state at the end of the N_i th

- Sub-head 3 is specified to be 10/12 Spartan Heavy (Video Technica Bold)—Sub-head 3 is a run-in side head and no space is provided above;
- A standard paragraph indention follows sub-heads;
- The running heads are set in 10 point Spartan Medium (Video Technica Medium) with ten points below—part title and the chapter title are used as right-hand and left-hand running heads respectively (neither synonym definition is made in the job specification; the definitions are made at the beginning of each part or chapter);
- The text block and footnote block bottom boundary procedures selected are defined as **Y1** and **Y3** respectively and the folio/running head procedure and the drop folio procedure are defined as **Y2** and **Y4** respectively;
- The control strings developed for chapter heading are defined to be synonyms **z3** through **z6** inclusive;
- In order to make use of the true italic, the synonym names **z8** and **z9** are defined to be Video Times Italic and Video Times Roman respectively; and finally,
- The style generator concludes the job specification.

Manuscript markup

As in the sample pages for "Strategy", the data tape for "MAINTAINABILITY" is logical unit one and the job specification of Figure 10.8 (and the definition of **z1** as the title of Part II) are inserted between lines one and two of the codes which appear at the top of page one of the manuscript, see Figure 10.9.

Figure 10.11
Style specification for *ACCENT: U.S.A.*



Scott, Foresman and Co.
Galaxy Series

Type Specification for ACCENT: U.S.A. L-73 (lino.) Poole

Trim: 6 3/4" x 9 1/8"

Type area: 31 x 46 picas

Page makeup: two 15 pica cols., 1 pica space; 48 lls. of 10/11-1/2 per col.

Back margin: 11/16"

Foot margin: 7/8" from trim to base of last line of text

Top margin: 9/16"

12 points space between last line of text and folio

Folios and running feet:

Folio: to be in outside margin
There is to be 12 points space between folio and running foot.

Running foot: There is to be 12 points space between last line of text and running foot.

Key to type specification for ACCENT: U.S.A.

A Text: 10/11-1/2 Times Roman, c/lc (roman, ital, bold, small caps as indicated on ms.)

15 pica cols. with 12 points space between cols.

B Headnotes: 10/11 Times Roman, c/lc, x 15 or x 25

C Title: 12 point Gothic #16, c/lc, x 15, flush left

D Author: 10 point Times Roman, ITALIC, flush left, under title, c/lc

E Footnotes and Credit lines: 8/9 Times Roman, x 15--24 pt. hairline rule flush left above text

F Running feet & Folios: 8 point Gothic #16, c/lc, 12 point space between running foot and folio
folio: * point Gothic #16 (to hang in margin)

The synonyms developed earlier are placed as shown in Figure 10.9. The fixed horizontal spaces placed between the letters in the word "CHAPTER" and between the letters of the words in the chapter title to effect thin spacing and letter spacing in the typeset sample are not shown on the manuscript.

The format label **F8** which introduces the credit line follows the word "POLICIES" without a word space and the asterisks are deleted from the manuscript because the format statements will generate them. After the footnote, the codes **[F9;Z7]** precede the first word of the chapter with no intervening wordspace.

The format labels **P1** and **P2** precede and follow the words of the center head with no intervening wordspaces. A wordspace at either point would put the center head off of true center by the amount of the wordspace.

The citation for white space required to accommodate Figure 2.1 appears in the second paragraph of page two of the manuscript. The figure, inclusive of its legends, requires 276 points; this value is assigned to **gv6** and the citation requesting the white space follows immediately.

The resulting sample pages are shown in Figure 10.10. Note that the processor allocated white space at the bottom of the text column. The citation appeared early enough that the space required was available on the page being composed and the amount of space requested was more than one-half text column.

A two-column sample

The two-column sample to be presented is a selection from *ACCENT: U.S.A.* published by Scott, Foresman and Company in 1964. This book displays a wide variety of typographic formats for the selection opening pages. Some employed a full, two-column title block followed by two-column text, others used part or all of one of the two columns, and others used the entire opening page for an illustration and the title.

Figure 10.12
Code strings and synonyms for
A Horseman in the Sky

Headnotes	[rb,360;ps,20;bl,22;tf,4;hn,1]
Title	[dn,68;ps,44;bl,60;tf,166]
Author	[dn,92;ps,20;bl,20;tf,14]
Center head	[[ne,cy,0[[eq,cx,0[[dj,35]];ne,cx,0[[dj,58]]]]; eq,cy,0[[eq,cx,0[[dj,6]];ne,cx,0[[dj,58]]]];ce] [dj,34;ju;hn,1]

Figure 10.13
Job specification for *ACCENT: U.S.A.*

Allowance for line numbers		[gv122,24;	(*)
Column measure		gv59,360;	15 picas—two columns
Gutter		gv57,24;	1 pica space
Nominal lines per column		gv60,48;	48 lines per column
Paragraph indention		gv76,1;	1 em
Typeface		gv41,4;	Times Roman†
Body of the book		gv11,20;gv26,23;	10/11½
Footnotes and credit lines	font/size	gv14,16;gv29,18;gv44,4;	8/9 Times Roman†
	font/size for citation	gv25,10;gv55,5;	(* 5 pt. Times Roman‡)
	text-footnote minimum space	gv172,12;	(* 6 pts.)
Running feet	specification of	gv123,0;	(*)
	font/size for folio	gv15,16;gv45,10;	8 pt. Gothic§
	folio width allotment	gv72,4;	(* 4 ems)
	left-hand text		Part title¶
	right-hand text		Selection title¶
	font/size for running foot	gv16,16;gv46,10;	8 pt. Gothic§
	space above	gv96,24]	12 pts. above
Text block *when bottom* procedure		[sy,Y1[[B7]];	(*)
Running foot		sy,Y2[[G7]];	(*)
Footnote block *when bottom* procedure		sy,Y3[[B6]];	(*)
Chapter opening folio		sy,Y4[[G7]]]	(*)
Synonyms for center head	start	[sy,U1[[ne,cy,0[[eq,cx,0[[dj,35]];ne,cx,0[[dj,58]]]]; eq,cy,0[[eq,cx,0[[dj,6]];ne,cx,0[[dj,58]]]];ce]];	(*)
	end	sy,U2[[dj,34;ju;hn,1]]]	(*)
Synonyms for true italic	to italic	[sy,Z8[[tf,14]];	(*)
	to roman	sy,Z9[[tf,4]]]	(*)
Style generator		[I9]	(*)
Half size code		[HS]	(*)

* These items are not a part of the conventional style specification.

† Use Video Times Roman, SR-II.
‡ Use Video Times Roman, SR-I.
§ Use Video Gothic, SR-II.

¶ Define synonyms at each part and selection opening, respectively.

For the present sample, the entire upper half of facing pages was devoted to the headnote, title, author line, and an illustration. The requirements, as reflected in the published book, pose an exercise in higher level, explicit mark-up using format statements. This sample also provides an example of the *half size* code. The style specification for *ACCENT: U.S.A.* is given in Fgure 10.11.

The format statements **B3** and **B4** will provide a full, two-column chapter heading followed by two column text. The requirement to have the text blocks on the second page (the facing page) be of the same depth and occupy the same location as those on the opening page is satisfied by a bit of trickery in manuscript markup. General variable 123 will be given the value one to provide the "running foot" format and the format statement **G7** will supply the hanging folio requirement. Though any of the center head format statement sets could be used for the simple heads required by this selection, job oriented synonyms are developed as a further example of Page–1 usage. Since selection title treatment varies widely in font, size, and spacing, explicit markup will be used to handle the headnote, title, and the author line.

Selection opening and center heads

The specification calls for headnotes to be set 10/11 Times Roman by 15 or 25 picas. In this instance, 15 picas will be used. The markup to prepare for headnotes sets right boundary to 360, font and size to

20/22 Video Times Roman, Size Range II (the *half size* code causes the processor to halve all size and distance measurements—20/22 with the *half size* code yields 10/11—and the size Range is II), and finally provides a one em space or indention before the first line of the headnote. This string is show in Figure 10.12.

The title is set each word on a separate line in 22 point Video Times Roman with 8 points between lines. The point size and body lead values are doubled; that is, 44/60. The sink below the headnote is one pica provided by a "down sixty-eight". The author's name is set 10 point Times Italic following a two pica sink from the last line of the title. The doubled values become 20 for point size and body lead and 92 for the sink. See Figure 10.12.

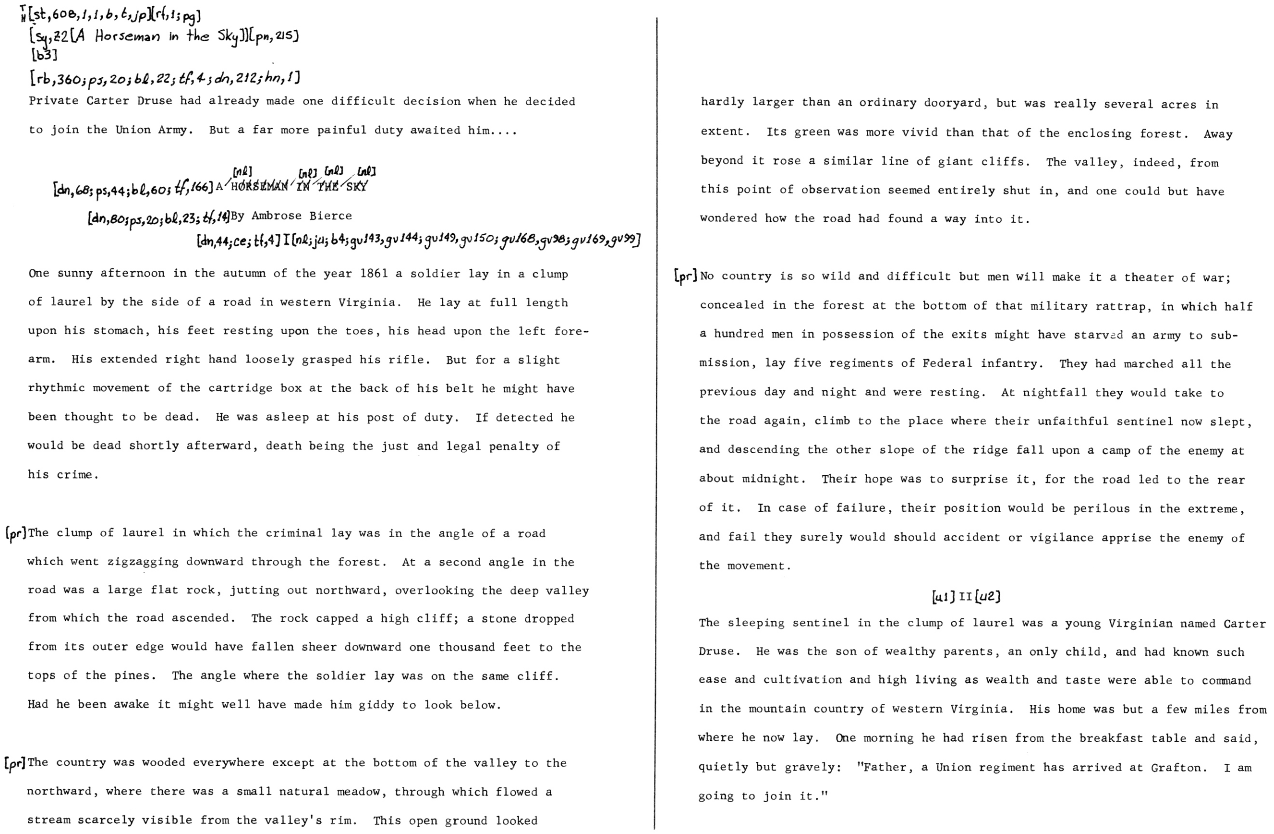
The selection is broken into three parts using centered Roman numerals. The code strings to provide for these are simplifications of the P-series of format statements. These strings, also shown in Figure 10.12, are given synonym names in the job specification.

The job specification

The job specification for "ACCENT: U.S.A." is shown in Figure 10.13; a comparison of the style specification and the job specification is detailed below:

- The job specification begins with an allowance of twelve points (doubled to 24 for use with *half size* code) for proof numbers;
- The column measure is fifteen picas—doubled to 360 points;
- The space between columns is one pica—doubled to 24 points;
- A nominal text column comprises 48 lines;

Figure 10.14
Manuscript for *A Horseman in the Sky*



TH[st,608,1,1,b,t,jp][rf,1;pg]
[sy,22[A Horseman in the Sky]][pn,215]
[b3]
[rb,360;ps,20;bl,22;tf,4;dn,212;hn,1]
Private Carter Druse had already made one difficult decision when he decided to join the Union Army. But a far more painful duty awaited him....

c/lc [dn,68;ps,44;bl,60;tf,166]A[nl]HORSEMAN[nl]IN[nl]THE[nl]SKY

[dn,80;ps,20;bl,23;tf,14]By Ambrose Bierce

[dn,44;ce;tf,4]I[nl;ju;b4;gv143,gv144;gv149,gv150;gv168,gv98;gv169,gv99]

One sunny afternoon in the autumn of the year 1861 a soldier lay in a clump of laurel by the side of a road in western Virginia. He lay at full length upon his stomach, his feet resting upon the toes, his head upon the left forearm. His extended right hand loosely grasped his rifle. But for a slight rhythmic movement of the cartridge box at the back of his belt he might have been thought to be dead. He was asleep at his post of duty. If detected he would be dead shortly afterward, death being the just and legal penalty of his crime.

[pr]The clump of laurel in which the criminal lay was in the angle of a road which went zigzagging downward through the forest. At a second angle in the road was a large flat rock, jutting out northward, overlooking the deep valley from which the road ascended. The rock capped a high cliff; a stone dropped from its outer edge would have fallen sheer downward one thousand feet to the tops of the pines. The angle where the soldier lay was on the same cliff. Had he been awake it might well have made him giddy to look below.

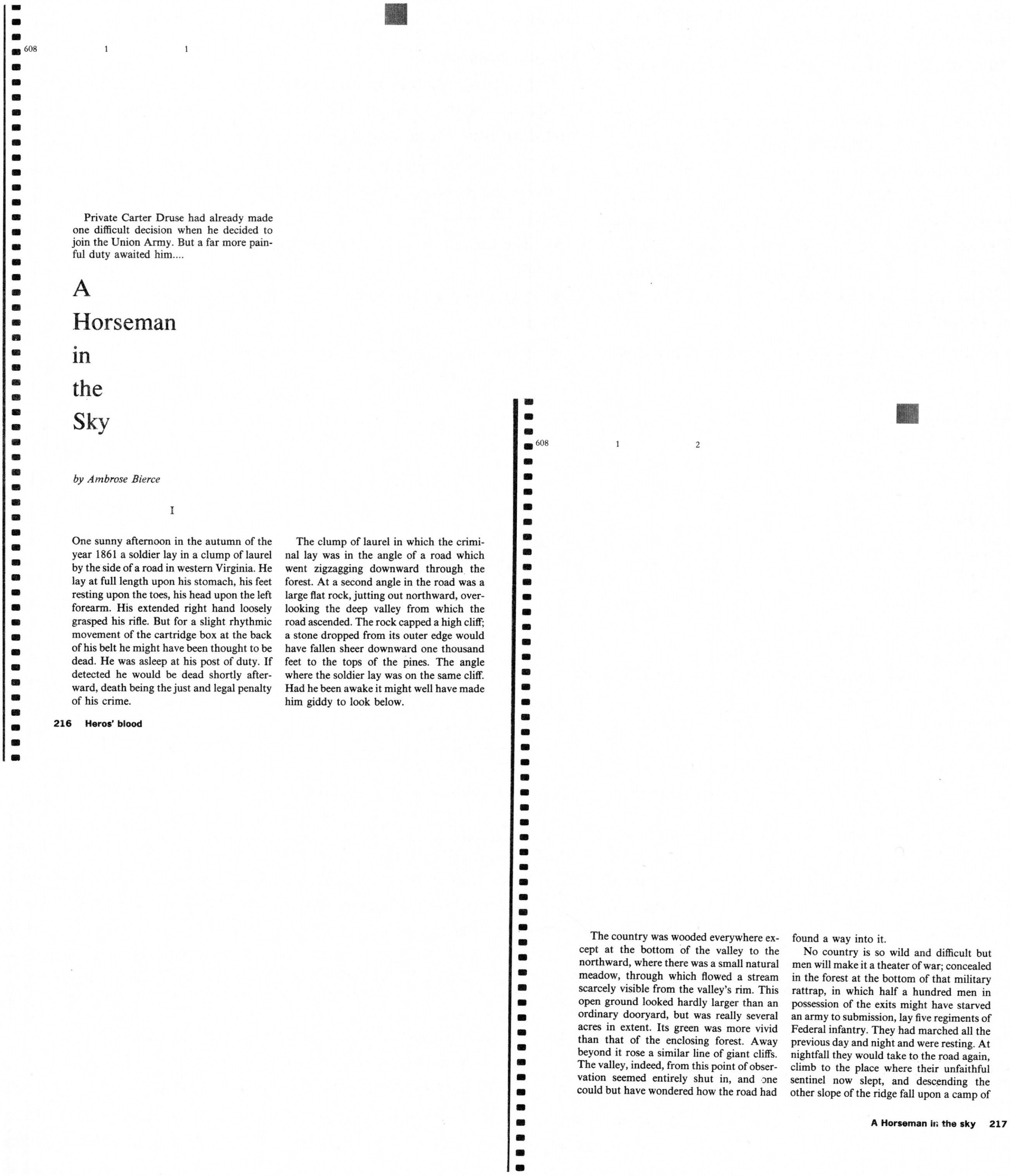
[pr]The country was wooded everywhere except at the bottom of the valley to the northward, where there was a small natural meadow, through which flowed a stream scarcely visible from the valley's rim. This open ground looked hardly larger than an ordinary dooryard, but was really several acres in extent. Its green was more vivid than that of the enclosing forest. Away beyond it rose a similar line of giant cliffs. The valley, indeed, from this point of observation seemed entirely shut in, and one could but have wondered how the road had found a way into it.

[pr]No country is so wild and difficult but men will make it a theater of war; concealed in the forest at the bottom of that military rattrap, in which half a hundred men in possession of the exits might have starved an army to submission, lay five regiments of Federal infantry. They had marched all the previous day and night and were resting. At nightfall they would take to the road again, climb to the place where their unfaithful sentinel now slept, and descending the other slope of the ridge fall upon a camp of the enemy at about midnight. Their hope was to surprise it, for the road led to the rear of it. In case of failure, their position would be perilous in the extreme, and fail they surely would should accident or vigilance apprise the enemy of the movement.

[u1]II[u2]

The sleeping sentinel in the clump of laurel was a young Virginian named Carter Druse. He was the son of wealthy parents, an only child, and had known such ease and cultivation and high living as wealth and taste were able to command in the mountain country of western Virginia. His home was but a few miles from where he now lay. One morning he had risen from the breakfast table and said, quietly but gravely: "Father, a Union regiment has arrived at Grafton. I am going to join it."

Figure 10.15
Sample pages from *A Horseman in the Sky*



608 1 1

Private Carter Druse had already made one difficult decision when he decided to join the Union Army. But a far more painful duty awaited him....

A Horseman in the Sky

by Ambrose Bierce

I

One sunny afternoon in the autumn of the year 1861 a soldier lay in a clump of laurel by the side of a road in western Virginia. He lay at full length upon his stomach, his feet resting upon the toes, his head upon the left forearm. His extended right hand loosely grasped his rifle. But for a slight rhythmic movement of the cartridge box at the back of his belt he might have been thought to be dead. He was asleep at his post of duty. If detected he would be dead shortly afterward, death being the just and legal penalty of his crime.

The clump of laurel in which the criminal lay was in the angle of a road which went zigzagging downward through the forest. At a second angle in the road was a large flat rock, jutting out northward, overlooking the deep valley from which the road ascended. The rock capped a high cliff; a stone dropped from its outer edge would have fallen sheer downward one thousand feet to the tops of the pines. The angle where the soldier lay was on the same cliff. Had he been awake it might well have made him giddy to look below.

216 Heros' blood

608 1 2

The country was wooded everywhere except at the bottom of the valley to the northward, where there was a small natural meadow, through which flowed a stream scarcely visible from the valley's rim. This open ground looked hardly larger than an ordinary dooryard, but was really several acres in extent. Its green was more vivid than that of the enclosing forest. Away beyond it rose a similar line of giant cliffs. The valley, indeed, from this point of observation seemed entirely shut in, and one could but have wondered how the road had found a way into it.

No country is so wild and difficult but men will make it a theater of war; concealed in the forest at the bottom of that military rattrap, in which half a hundred men in possession of the exits might have starved an army to submission, lay five regiments of Federal infantry. They had marched all the previous day and night and were resting. At nightfall they would take to the road again, climb to the place where their unfaithful sentinel now slept, and descending the other slope of the ridge fall upon a camp of

A Horseman in the sky 217

- The standard paragraph indention is em;
- The typeface for the body of the book is Video Times Roman, Size Range II which contains 10 point;
- The body of the book is set 10/11½—doubled to 20/23;
- Footnotes and credit lines are set 8/9 (Video Times Roman, Size Range II)—doubled to 16/18—they are separated from text by six points—doubled to 12;
- The folio and running feet are set eight point Video Gothic, Size Range II, and are separated from text by 12 points; part title and selection title are used as left-hand and right-hand running feet respectively (neither synonym definition is made in the job specification; the definitions are made at the beginning of each part and selection);
- The text block and footnote block bottom boundary procedures selected are defined as **Y1** and **Y3** respectively and the folio/running foot procedures (the opening page "drop folio" is also a running foot in this job) are defined as **Y2** and **Y4** respectively;
- The control strings for center heads are defined to be the synonyms **U1** and **U2**;
- In order to make use of true italic, the synonym names **Z8** and **Z9** are defined to be Video Times Italic and Video Times Roman respectively; and finally,
- The two-column style generator and the *half size* code (which must precede any typeset material—either explicit text or generated text) conclude the style specification.

Manuscript markup

As before, the job specification and the definition of **Z1** appear on the data tape between the codes which appear on lines one and two of the manuscript shown in Figure 10.14. The codes developed earlier to govern typesetting of the chapter heading are written on the manuscript as shown.

White space (of the depth required by the chapter opening) must be reserved at the top of the second page to allow for subsequent insertion of a four-column illustration. Knowledge of the working of **B7** provides a way to accomplish this requirement. The format statement **B7** is the text block *when bottom* procedure; it initializes the attributes of the page to follow. The Y-coordinates of the text blocks are the values assigned to general variables 143 and 149 and the bottom boundaries of the text blocks are the values assigned to general variables 168 and 169.

After the chapter headings have been set, the format statement **B4** will compute block origins and bottom boundaries for the text blocks on the chapter opening page. The general variables used in these computations (and subsequent assignment of attributes for the chapter opening page) are general variables 144 and 150 for block origins and general variables 98 and 99 for bottom boundaries. The general variables which initialize the second page are assigned the values computed by **B4**. These assignments are shown on the first page of the manuscript in Figure 10.14 immediately preceding the opening of text.

The sample pages are shown in Figure 10.15.

A PAGE-1 dictionary

Appendix A

Each dictionary entry includes a statement of the legal form(s) in which the code may be used. The symbols used in these statements are defined as follows:

⟨x⟩	the thing called x
::=	is defined to be
|	exclusive or
$\{z\}_i^j$	z is to be repeated at least i times but not more than j times. When i is omitted, its value is assumed one.

The terms *parameter, constant,* and *variable* (which were defined in Part II) are repeated here using these symbols.

⟨parameter⟩ ::= ⟨constant⟩|⟨variable⟩

⟨constant⟩ ::= $\{\langle\text{digit}\rangle\}_1^9$
⟨digit⟩ ::= **1|2|3|4|5|6|7|8|9|0**

⟨variable⟩ ::= ⟨global variable⟩|⟨typographic variable⟩|⟨system variable⟩
⟨global variable⟩ ::= **gv**⟨number⟩|**iv**⟨number⟩|**fn**|**pi**|**pn**
⟨typographic variable⟩ ::= **bb**|**bl**|**bs**|**lb**|**mi**|**mx**|**pd**|**ps**|**pw**|**rb**|**sw**|**tb**|**tf**
⟨system variable⟩ ::= **cc**|**cx**|**cy**

A PAGE-1 dictionary

aa — *alter all* — alteration operator

The alter statement takes the form

.... [aa,⟨line⟩,⟨word space⟩-⟨line⟩,⟨word space⟩[⟨string⟩]]

Provides the capability to substitute new for existing material in a composed master file. The material to be deleted is both text and control. The *alter all* statement is used only after a job has been composed and a master file exists—the alteration is written against the master file. The string in the *alter all* statement carries the material which is to replace original material in the master file. The material to be replaced is that which follows the first line and word space cited to and including the second line and word space cited. The first interword space in a line is word space one. An alteration statement which replaces only the first word in a line may be "short handed" by stating the line number and word space zero; that is, [aa,⟨line⟩,0[⟨string⟩]]. A statement which changes one entire line may state only the line number: [aa,⟨line⟩[⟨string⟩]]. The statement [aa,2-3[⟨string⟩]] will substitute the string supplied for lines two and three of the composed master file.

Notes: (1) The **lu** function code establishes the logical unit of the job against which an alteration statement or a series of alteration statements is to apply. (2) The logical unit specified in the start take header is assumed at the beginning of an alteration take. (3) Em, en and thin spaces and space provided by the **hn** function code are not considered to be word spaces for purposes of counting word spaces in alteration. (4) The PAGE–1 Composition System assumes that the word space which follows a word is attached to that word. Care must be exercised when replacing a word to replace the following word space also.

ab — *at bottom* — boundary operator

The boundary procedure takes the form .... [ab[⟨string⟩]]

Provides the capability to retrieve and to execute a string of control codes—or a string composed of control codes and text—at the first bottom boundary encounter after the procedure is placed in force. The *at bottom* procedure is executed once and then discarded. If *at bottom* and *when bottom* procedures are both present, the *at bottom* procedure takes precedence. The *at bottom* procedure will be executed at the first bottom boundary encounter; the *when bottom* procedure will be executed at the second and subsequent bottom boundary encounters. If bottom boundary and right boundary procedures are used together, the precedence of execution is established by the order of their first entry into input. For example, in the control string fragment [.....ar[[x1]];ab[[x2]];.....], the right boundary procedures will take precedence over the bottom boundary procedures throughout the entire job.

Notes: (1) One boundary string must be completed before another is started. For example, if right boundary procedures are to take precedence over bottom boundary procedures when they appear together, care must be exercised to avoid the circumstance in which the bottom boundary is encountered during execution of the right boundary procedure. The use of a **dn**, **dj**, **nl**, or a **pr** within the right boundary procedure *may* result in a bottom boundary encounter—thereby calling the bottom boundary procedure—and composition will terminate. (2) An *at bottom* procedure may be placed in force by simply stating the desired procedure explicitly at any point in the manuscript or it may be included in any of the four boundary procedures. (3) If text is to be set by an *at bottom* procedure, the text must be preceded by a *start block* command or by the redefinition of the bottom boundary to make the block deeper.

ac — *alter control* — alteration operator

The alter statement takes the form

.... [ac,⟨line⟩,⟨word space⟩-⟨line⟩,⟨word space⟩[⟨string⟩]]

Provides the capability to substitute new for existing material in a composed master file. The material to be deleted is control only. The *alter control* statement is used only after a job has been composed and a master file exists—the alteration is written against the master file. The string in the *alter control* statement carries the material which is to replace original material in the master file. The material to be replaced is that which follows the first line and word space cited to and including the second line and word space cited. The first interword space in a line is word space one. An alteration statement which replaces only the controls associated with the first word in a line may be "short handed" by stating the line number and word space zero; that is, [ac,⟨line⟩,0[⟨string⟩]].

Notes: (1) The **lu** function code establishes the logical unit of the job against which an alteration statement or a series of alteration statements is to apply. (2) The logical unit specified in the start take header is assumed at the beginning of an alteration take. (3) Em, en and thin spaces and space provided by the **hn** function code are not considered to be word spaces for purposes of counting word spaces in alteration.

ad — *add* — arithmetic operator

The arithmetic statement
takes the form ....[ad,⟨variable⟩,⟨parameter⟩,⟨parameter⟩]

Provides the capability to do addition. The variable is assigned the numeric (constant) value which accrues when the parameters are added. The first parameter is the augend; the second is the addend. The parameters may be numeric constants or they may be variables—the arithmetic operation [ad,gv90,cy,bl] will add the values of the typographic variables **cy** and **bl** at the time the operation is executed and assign the result to **gv90**.

ar — *at right* — boundary operator

The boundary procedure takes the form .... [ar[⟨string⟩]]

Provides the capability to retrieve and to execute a string of control codes—or a string composed of control codes and text —at the first subsequent right boundary encounter after the procedure is placed in force. The *at right* procedure is executed once and then discarded. If *at right* and *when right* procedures are both present, the *at right* procedure takes precedence. The *at right* procedure will be executed at the first right boundary encounter; the *when right* procedure will be executed at the second and subsequent right boundary encounters. If bottom boundary and right boundary procedures are used together, the precedence of execution is established by the order of their first entry into input. For example, in the control string fragment [.....ar[[x1]];ab[[x2]];.....], the right boundary procedures will take precedence over the bottom boundary procedures throughout the entire job.

Notes: (1) One boundary string must be completed before another is started. For example, if right boundary procedures are to take precedence over bottom boundary procedures when they appear together, care must be exercised to avoid the circumstance in which the bottom boundary is encountered during execution of the right boundary procedure. The use of a **dn, dj, nl,** or a **pr** within the right boundary procedure *may* result in a bottom boundary encounter—thereby calling the bottom boundary procedure—and composition will terminate. (2) An *at right* procedure may be placed in force by simply stating the desired procedure explicitly at any point in the manuscript or it may be included in any of the four boundary procedures. (3) Before execution of the *at right* string, a new line is executed; **cy** is incremented appropriately and **cx** is set to zero.

at — *alter text* — alteration operator

The alter statement takes the form
..............[at,⟨line⟩,⟨word space⟩-⟨line⟩,⟨word space⟩[⟨string⟩]]

Provides the capability to substitute new for existing material in a composed master file. The material to be deleted is text only. The *alter text* statement is used only after a job has been composed and a master file exists—the alteration is written against the master file. The string in the *alter text* statement carries the material which is to replace original material in the master file. The material to be replaced is that which follows the first line and word space cited to and including the second line and word space cited. The first interword space in a line is word space one. An alteration statement which replaces only the first text word in a line may be "short handed" by stating the line number and word space zero; that is, [at,⟨line⟩,0[⟨string⟩]]. A statement which changes one entire line may state only the line number: [at,⟨line⟩[⟨string⟩]]. The statement [at,2-3[⟨string⟩]] will substitute the string supplied for lines two and three of the composed master file.

Notes: (1) The **lu** function code establishes the logical unit of the job against which an alteration statement or a series of alteration statements is to apply. (2) The logical unit specified in the start take header is assumed at the beginning of an alteration take. (3) Em, en and thin spaces and space provided by the **hn** function code are not considered to be word spaces for purposes of counting word spaces in alteration. (4) The PAGE–1 Composition System assumes that the word space which follows a word is attached to that word. Care must be exercised when replacing a word to replace the following word space also.

bb — *bottom boundary* — typographic variable

The assign statement takes the form ....[bb,⟨parameter⟩]

Establishes the *bottom boundary* to be in force and allows direct involvement of *bottom boundary* in arithmetic operations or conditional procedures. The bottom boundary is expressed in points below the block origin. A statement which will change the bottom boundary may be inserted at any point; it will take effect if or when **cx** is equal to zero. The value of the **bb** typographic variable is updated immediately but the boundary change is not effective until current line termination. The parameter may be a numeric constant or a variable. The statement [bb,456] will cause the bottom boundary to be set 456 points (38 twelve point lines) below the block origin. The statement [bb,gv89] will cause the bottom boundary to be set below block origin by the number of points assigned to **gv89** at the time the statement is executed. Changing the value of **gv89** subsequent to the execution of [bb,gv89] will not in itself change the bottom boundary .

Notes: (1) If no bottom boundary procedure is in force. bottom boundary encounter has no effect; typesetting will continue subject only to subsequent start block or new print region commands or to the limitations noted in (3) and (4) below. (2) If more text is to be set in a block during execution of its bottom boundary procedure, the bottom boundary must be extended before the text can be set. The attempt to set more text will otherwise cause another bottom boundary encounter and composition will terminate. (3) The bottom boundary is initialized to 1584 points at system start; this value is based on VideoComp film processing characteristics. (4) If a composition job is for proof purposes (ie., the VideoComp output will be on stabilization paper), the effective bottom boundary—between a **by** of zero and the last possible line of text before the cut is 1208 points.

bk — *backspace* — function code

The backspace statement takes the form
......[bk,{⟨parameter⟩|⟨parameter⟩,⟨parameter⟩|,⟨parameter⟩}$^{1}_{1}$]

Provides the capability to specify a fixed *backspace*. The command may have one or two parameters; the total amount the backspace provided is the sum of the first parameter (interpreted in ems) and the second parameter (interpreted in VIDs). The statement [bk,2] provides a two em backspace, the statement [bk,2,25] provides a two and one-half em backspace, and the statement [bk,,25] provides a one-half em (one en) backspace.

Notes: (1) An em is a space of horizontal length equal to the set width in force; a VID is 1/50th of an em. (2) The left boundary may not be violated by the **bk** code; if the amount of the backspace would dictate crossing the left boundary, the backspace provided will extend only to the left boundary.

bl — *body lead* — typographic variable

The assign statement takes the form .... [bl,⟨parameter⟩]

Provides the capability to specify or change the *body lead* to be in force and allows direct involvement of body lead in arithmetic operations or conditional procedures. The parameter may be a numeric constant or a variable; the statement [bl,12] specifies body lead to be twelve or the statement [bl,gv26] specifies body lead to be the value of **gv26** at the time the statement is executed. Changing the value of **gv26** subsequent to the execution of [bl,gv26] does not in itself change the body lead.

Notes: (1) At system start, body lead is initialized to ten. (2) Interlinear space or leading is computed by subtracting point size from body lead. The values of **bl** and **ps** used in the computation are the largest values that have been in force in the interval between termination of the last complete line set and the present termination. Computed leading is added to those lines terminated by a natural line break or by the **nl** or **pr** codes. (Except that leading is not applied to the last line in a block when the **qc**, **qd**, **jd**, or **jv** commands are in force. (3) The PAGE–1 Composition System will provide negative lead capability via point size and body lead combinations; but the net resulting film advance must be positive.

bs — *basic interword space* — typographic variable

The assign statement takes the form .... [bs,⟨parameter⟩]

Provides the capability to alter the *basic interword space* during composition. The basic interword space is that space to be provided between words in the typesetting of unjustified material (eg., flush left) and to be tentatively assigned—subject to adjustment between limits in justification—in the typesetting of justified material. The basic interword space is specified in relative units; the relative unit, for VideoFonts supplied by RCA Graphic Systems, is related to the size range of the font. Recommended values for the interword spaces (minimum, basic, and maximum) are specified for and supplied with each digital font. When fonts are loaded on the system disc, these recommended (or perhaps other) interword spaces are loaded also. When a font is specified for use in a job, these stored, "standard" values are used to set the minimum, basic, and maximum interword space typographic variables. Should values different from the "system standard" values be desired in a given job, care must be exercised that the desired values are reinserted each time (after) the font is specified.

bx — *block origin–horizontal* — function code

The block origin
statement takes the form ....[bx,⟨parameter⟩,⟨parameter⟩]

Specifies the X-coordinate of the origin of the block cited by the first parameter to be rightward from the left edge of the print region by the number of points stated in the second parameter. Block origin may be specified once at the beginning of the job. Unless the requirements of the job dictate changes in block location during composition, **bx** need not be stated again. Or it may be specified—perhaps changed several times—on each page; the value assigned to **bx** at the time the print region is closed, is the value for that page.

Notes: (1) At system start, **bx**s for all 32 blocks are initialized to zero. (2) If line numbers are desired for proof mode, a twelve point allowance should be made at the left edge of the print region.

by — *block origin–vertical* — function code

The block origin
statement takes the form ....[by,⟨parameter⟩,⟨parameter⟩]

Specifies the Y-coordinate of the origin of the block cited by the first parameter to be below the top edge of the print region by the number of points stated in the second parameter. Block origin may be specified once at the beginning of the job. Unless the requirements of the job dictate changes in block location during composition, **by** need not be stated again. Or it may be specified—perhaps changed several times—on each page; the value assigned to **by** at the time the print region is closed, is the value for that page.

Note: At system start, **by**s for all blocks are reset to zero.

cb — *continue block* — function code

The continue block statement takes the form .... [cb]

Instructs the system that typesetting is to be resumed in the primary block starting at the point at which that block was interrupted earlier. The primary block must have been interrupted earlier for this code to have meaning; **cb** will be ignored if the system is not in interrupt block mode. The interrupting block will itself be interrupted—its environment saved—in order that it may be subsequently re-entered by another interrupt block statement. See **ib** and **sb** for further discussion.

cc — *current character* — datum

The assign statement takes the form .... [⟨variable⟩,cc]

Provides the capability to retrieve the most recently set character for use as a parameter in a conditional procedure. The code has its primary use in a boundary procedure of the form [wr[[eq,cc,⟨text character⟩[⟨string⟩]] in which a string is to be executed only if the final character on a line is a specific character. In the example, if the line has been hyphenated, the hyphen will be the **cc**; this feature makes it possible to recognize hyphenated lines. The text character against which **cc** is to be compared in the procedure must be entered as the decimal equivalent of the internal code for the character.

Note: If **cc** is to be used to test characters within a line, the instruction identifying the test must follow each character. This is necessary since the action of **cc** will function only once for each time it is called.

ce — *center horizontally* — function code

The center horizontally statement takes the form .... [ce]

Establishes a typesetting mode in which text material is set unjustified. The words in a line are separated by the basic interword space; each time a word which would overset the right boundary is encountered, the line is terminated at the preceding interword space and unused space is inserted half at the beginning and half at the end of the line. Center horizontally mode provides a "ragged left and right" output column. The center horizontally mode is cancelled by **ql**, **qr**, or **ju** mode.

cn — *consider* — function code

The consider statement takes the form .... [cn]

The *ignore* and *consider* function codes provide the capability to include or omit material from a manuscript in order to generate

.....*Continued overleaf*

two output versions from a single manuscript. For example, the input:

.....preceding text [ne,gv1,1[[ig]]]conditional text
[cn]following text.....

will produce only preceding text and following text if **gv1** is equal to one but will produce preceding text, conditional text, and following text if **gv1** is equal to something other than one. The **cn** cannot, however, be conditional since, if the processor is in ignore mode, the conditional statement will not be executed.

Notes: (1) At system start, the ignore-consider condition is initialized to **cn**. (2) Both text and control following an **ig** code will be ignored until a **cn** code is encountered. (3) If the *ignore* code appears in any of the four boundary procedures, in a saved string, or in a defined string (for use with the *execute defined string* code), the *consider* code must appear before the end of the same procedure or string. (4) The *ignore* code need not precede every occurrance of the *consider* code; if the processor is in *consider* mode, the appearance of a *consider* code will have no effect.

cv — *center vertically* — function code

The center vertically statement takes the form .... [cv]

Causes all lines which have been set in a block to be centered vertically between the top and bottom boundaries of the block without expansion of interlinear space. The **cv** code is executed when the block in which it is in force is terminated and is then discarded; if it is desired on succeeding pages, it may be put in the bottom boundary procedure. Surplus space between the last line set—excluding leading on the last line—and the bottom boundary is allocated one-half above and one-half below the text. The division of space is in increments of one point for the 70|820 and in increments of 1/32nd point for the 70|830.

cx — *current horizontal setting position* — datum

The assign statement takes the form .... [⟨variable⟩,cx]

Provides the capability to retrieve the *current horizontal setting position* and to involve it in arithmetic statements, conditional and boundary procedures, and as a parameter to function codes. The value of **cx** is measured from the left boundary to the right edge of the last character set; it is expressed in points. The system maintains horizontal setting position in internal units; when **cx** is used, the result of the conversion to points is rounded up.

Note: (1) At system start, **cx** is initialized to zero; it is subsequently reset to zero on execution of natural line breaks (prior to execution of *at right* or *when right* procedures which may be in force) and forced line breaks.

cy — *current vertical setting position* — datum

The assign statement takes the form .... [⟨variable⟩,cy]

Provides the capability to retrieve the *current vertical setting position* and to involve it in arithmetic statements, conditional and boundary procedures, and as a parameter to function codes.

Notes: (1) At system start, **cy** is initialized to zero. (2) The line-by-line update of **cy** takes two forms depending on the method by which a given line was terminated. The termination of one line and the initiation of the following line may come about in two ways: first, by forced or natural new line and second, by a sink command. The new line class includes forced line breaks by **nl** or **pr** codes and natural line breaks which result from right boundary encounter. The sink class includes the forced line breaks specified by down (either **dj** or **dn**) commands. Both of the down commands operate the same for purposes of **cy** update during composition—the **dj** parameter is nominal only and if **jd** is in force when the block is closed, the amount of space provided may be increased.

If line termination and new line initiation are the result of a forced or a natural line break, the update of **cy** is as follows: first, interlinear leading is computed as the difference between **bl** and **ps** using the largest values of **bl** and **ps** defined after termination of the last complete line set and the present termination (if no specification of **bl** or **ps** was made in that interval, the most recently specified values will be used in the computation); next, this interlinear space is added to **cy**; then, if the **nl** or **pr** code which caused a forced line break has a parameter, the parameter is multiplied by the value of **bl** at the termination, and this value is also added to **cy**; finally, the largest value of **ps** on the line being terminated is added to **cy**. The updated value of **cy** after update extends to the bottom edge—including leading—of the line just terminated plus any blank lines specified.

If line termination and new line initiation are the result of a sink command, the update of **cy** is as follows: the value supplied by the parameter to the sink command is added to **cy**; then the initial value of **ps** for the line being initiated is added to **cy**. The updated value of **cy** then extends to the baseline of the new line. If this "sink initiated" line is in turn terminated by a natural or forced line break, point size will *not* be added to **cy** in the update —it has already been accounted for.

dc — *set decimal number* — function code

The set decimal number
statement takes the form .... [dc,⟨parameter⟩]

Provides the capability to set as an arabic numeral the decimal equivalent of any constant or variable. For example, the statement [dc,pn] will set the value stored in the *page number counter* at the time the statement is executed. (The page number counter must be incremented separately—perhaps in the bottom boundary procedure—by means of the arithmetic statement.)

df — *difference* — arithmetic operator

The arithmetic statement
takes the form .... [df,⟨variable⟩,⟨parameter⟩,⟨parameter⟩]

Provides the capability to do subtraction. The first parameter is the minuend; the second is the subtrahend. The variable is assigned the numeric (constant) value which accrues when the second parameter is subtracted from the first. No notice of sign is given to the user. The parameters may be numeric constants or they may be variables—the arithmetic operation [df,gv127,pd,cy] will subtract the value of the typographic variable **cy** from the value of the typographic variable **pd** at the time the operation is executed and assign the result to **gv127**.

dj — *justifiable down* — function code

The sink statement takes the form .... [dj,⟨parameter⟩]

A forced line break; causes the current line to be terminated and a new line to be initiated starting at the left boundary and on a baseline which is below the current baseline by a nominal vertical sink specified in points. When the code is executed, the vertical sink provided is that specified by the parameter. If the

jd command is present when the block is terminated, all excess space between the baseline of the last line in the block and the bottom boundary will be divided among the justifiable down lines in the block. If there are no justifiable down lines in a block in which the **jd** command is in force, the **jd** will have no effect. See **cy** for effect of **dj** on the vertical setting position.

dn — *fixed down* — function code

The sink statement takes the form ....[dn,⟨parameter⟩]

A forced line break; causes the current line to be terminated and a new line to be initiated starting at the left boundary and on a baseline which is below the current baseline by a fixed vertical sink specified in points. See **cy** for the effect of **dn** on the current vertical setting position.

dw — *baseline down* — function code

The baseline down statement takes the form .... [dw,⟨parameter⟩]

Provides the capability to control the VideoComp baseline. The amount by which the baseline is to be lowered from the zero, or nominal, position is specified by the parameter in increments of 1/32nd point. If the output device is the 70|820, the **dw** must be specified in increments of one-half point; that is, the parameter must be a multiple of sixteen. Either [dw,0] or [up,0] may be used to restore the baseline to nominal.

Notes: (1) The parameter specifies an absolute displacement from the nominal baseline position. (2) Current vertical setting position, **cy**, is not affected by a **dw** command. (3) At system start, **dw** is reset to zero; that is, baseline position is nominal. (4) The VideoComp aperture limits maximum downward beam displacement for the 70|820 to 18 points and for the 70|830 to 31 and 31/32 points. (5) If a baseline down for the 70|820 is not specified in one-half point increments, the parameter supplied will be rounded up to the next multiple of sixteen.

ea — *eliminate all* — alteration operator

The eliminate statement takes the form

.... [ea,⟨line⟩,⟨word space⟩-⟨line⟩,⟨word space⟩]

Provides the capability to delete material (both text and control) from a composed master file. The *eliminate all* statement is used only after a job has been composed and a master file exists—the alteration is written against the master file. The material to be deleted is that which follows the first line and word space cited to and including the second line and word space cited. The first interword space in a line is word space one. An alteration statement which deletes only the first word in a line may be "short handed" by stating the line number and word space zero; that is, [ea,⟨line⟩,0]. A statement which deletes an entire line may state only the line number: [ea,⟨line⟩].

Notes: (1) The **lu** function code establishes the logical unit of the job against which an alteration statement or a series of alteration statements is to apply. (2) The logical unit specified in the start take header is assumed at the beginning of an alteration take. (3) Em, en and thin spaces and space provided by the **hn** function code are not considered to be word spaces for purposes of counting word spaces in alteration.

ec — *eliminate control* — alteration operator

The eliminate statement takes the form

.... [ec,⟨line⟩,⟨word space⟩-⟨line⟩,⟨word space⟩]

Provides the capability to delete material (only control) from a composed master file. The *eliminate control* statement is used only after a job has been composed and a master file exists—the alteration is written against the master file. The material to be deleted is that which follows the first line and word space cited to and including the second line and word space cited. The first interword space in a line is word space one. An alteration statement which deletes only the controls associated with the first word in a line may be "short handed" by stating the line number and word space zero; that is, [ec,⟨line⟩,0]. A statement which deletes only the controles associated with a single line may state only the line number: [ec,⟨line⟩].

Notes: (1) The **lu** function code establishes the logical unit of the job against which an alteration statement or a series of alteration statements is to apply. (2) The logical unit specified in the start take header is assumed at the beginning of an alteration take. (3) Em, en and thin spaces and space provided by the **hn** function code are not considered to be word spaces for purposes of counting word spaces in alteration.

eq — *equal* — conditional operator

The conditional procedure
takes the form .... [eq,⟨parameter⟩,⟨parameter⟩[⟨string⟩]]

Provides the capability to execute strings of control codes or to set strings of text or to handle any combination of control codes and text on the condition that two parameters are *equal.* If the condition is not fulfilled, the string is not executed. For example, the conditional procedure [eq,gv199,0[[ql]]] will, if the value of the variable **gv199** is *equal* to the numeric constant zero at the time the procedure is executed, establish the flush left mode. If not, the justification mode is in force will remain in force.

es — *execute saved string* — function code

The execute saved string
statement takes the form .... [es,⟨parameter⟩]

Provides the capability to retrieve and to execute either of two saved strings which have been created by the *mark begin* and *mark end* function codes. The parameter may be a constant or a variable; but in either case its value must be one or two. The parameter designates which of the two saved strings is to be executed. See **mb** and **me** for the creation of saved strings.

et — *eliminate text* — alteration operator

The eliminate statement takes the form

...............................[et,⟨line⟩,⟨word space⟩-⟨line⟩,⟨word space⟩]

Provides the capability to delete material (only text) from a composed master file. The *eliminate text* statement is used only after a job has been composed and a master file exists—the alteration is written against the master file. The material to be deleted is that which follows the first line and word space cited to and including the second line and word space cited. The first interword space in a line is word space one. An alteration statement which deletes the text word in a line may be "short handed" by stating only the first line number and word space zero; that is, [et,⟨line⟩,0]. A statement which deletes all of the text in a single line may state only the line number: [et,⟨line⟩].

Notes: (1) The **lu** function code establishes the logical unit of the job against which an alteration statement or a series of alteration statements is to apply. (2) The logical unit specified in the start take header is assumed at the beginning of an alteration take. (3) Em, en and thin spaces and space provided by the **hn** function code are not considered to be word spaces for purposes of counting word spaces in alteration.

ex — *execute defined string* — function code

The execute defined string statement takes the form .... [ex,⟨parameter⟩[⟨string⟩]]

Provides for execution of a format statement, a synonym, or an explicitly stated string a specified number of times. The parameter specifies the number of times the string is to be executed. Provision to alter the number of times the defined string is executed after execution has started, is provides by **gv0**. General variable zero may be set—in a right boundary procedure, for example—to a value corresponding to an altered number of executions. The value assigned to **gv0** specifies a number of executions in addition to those already executed.

fl — *fill to measure* — function code

The fill statement takes the form .... [fl{,⟨parameter⟩}$^{2}_{0}$]

Provides the capability to "leader out" a line with a text character or text character pair or to flush two components of a line to opposite ends of the line. When the code is used with only one parameter, the system will assume the second parameter to be the same as the first. In use, the *fill to measure* statement is placed in the text in the position the leader is to occupy and the end of the line is specified by a forced line break.

The code works by computing the surplus space on the line and inserting a number of text character pairs sufficient to fill out the line. When the fill has no text on either side, space resulting from the fact that an integral number of text character pairs does not exactly fit is placed in front of the leader. When there is text on both sides of the fill, the surplus space is allocated to the left and the right so that the fill characters begin at a point which is a multiple of the fill character-pair width from the right boundary. Up to five *fill to measure* commands with the same or different fill characters may be executed on a line or between tab stops. The space to be filled is divided equally among the individual fills.

Note: In using the *fill to measure* code (with em or en dashes) to generate horizontal rules, care must be exercised to choose a point size that is an even numbered submultiple of measure.

fn — *footnote citation counter* — global variable

The assign statement takes the form .... [fn,⟨parameter⟩]

Provides the capability to specify, manipulate, and typeset the footnote citation. The global variable **fn** is actually a dedicated general variable. The footnote citation counter operates on the basis that footnotes citations are arabic numerals. (Footnote citations must be set from the basic subset zero when the **fn** global variable is used.) The footnote citation counter may be initialized to one via the assign statement [fn,1] and incremented appropriately with the arithmetic statement [ad,fn,fn,1]. It may then be typeset at the desired time using the set decimal number code [dc,fn]. At system start **fn** is initialized to zero.

ge — *greater than or equal* — conditional operator

The conditional procedure takes the form .... [ge,⟨parameter⟩,⟨parameter⟩[⟨string⟩]]

Provides the capability to execute strings of control codes or to set strings of text or to handle any combination of control codes and text on the condition that the first parameter is *greater than or equal* to the second parameter. If the condition is not fulfilled, the string is not executed. For example, consider the conditional procedure [ge,gv70,gv82[[df,gv84,gv89,gv70]]]; it will, if the value of the variable **gv70** is *greater than or equal* to the value of the variable **gv82** at the time the procedure is executed, subtract the value of **gv70** from that of **gv89** and assign the result to **gv84**. If not, the value of **gv84** will not be altered by the procedure.

gr — *greater of* — arithmetic operator

The arithmetic statement takes the form .... [gr,⟨variable⟩,⟨parameter⟩,⟨parameter⟩]

Provides the capability to do a comparison of values and to store the greater. If the two parameters are equal, their common value is assigned to the variable. The parameters may be numeric constants or they may be variables—the arithmetic operation [gr,gv86,gv85,gv84]— will compare the values of **gv85** and **gv84** at the time the operation is executed and assign the value of the greater to **gv86**.

gt — *greater than* — conditional operator

The conditional procedure takes the form .... [gt,⟨parameter⟩,⟨parameter⟩[⟨string⟩]]

Provides the capability to execute strings of control codes or to set strings of text or to handle any combination of control codes and text on the condition that the first parameter is *greater than* the second parameter. If the condition is not fulfilled, the string is not executed. For example, the conditional procedure [gt, ps,10[[ps,10]]] will, if the point size in force at the time the procedure is executed is *greater than* ten, set the point size to ten. If not, point size will not be altered by the procedure.

gv — *general variable* — global variable

The assign statement takes the form .... [gv⟨number⟩,⟨parameter⟩]

Provides the capability to establish specific values for meaningful quantities and subsequently to involve them in arithmetic statements and conditional procedures. The system provides for up to 200 general variables numbered one through 200 inclusive. General variable zero is an additional **gv** dedicated to the *execute defined string* statement. The parameter in the assign statement may be a numeric constant or a variable; the statement [gv11,10] establishes the value of **gv11** to be ten or the statement [gv192,cy] establishes the value of **gv192** to be equal to the value of the typographic variable **cy** at the time the statement is executed.

Notes: (1) At system start, all general variables are set to zero. (2) A general variable may not be assigned a value greater than 32767.

hn — *horizontal space* — function code

The horizontal space statement takes the form

.... [hn,{⟨parameter⟩|⟨parameter⟩,⟨parameter⟩|,⟨parameter⟩}$^{1}_{1}$]

Provides the capability to specify a fixed *horizontal space*. The command may have one or two parameters; the total amount the horizontal space provided is the sum of the first parameter (interpreted in ems) and the second parameter (interpreted in VIDS). The statement [hn,2] provides a two em horizontal space, the statement [hn,2,25] provides a two and one-half em horizontal space, and the statement [hn,,25] provides a one-half em (one en) horizontal space.

Notes: (1) An em is a space of horizontal length equal to the set width in force; a VID is 1/50th of an em. (2) Horizontal space provided by the **hn** code is not altered in justification and it is not counted as a word space for purposes of alteration.

hs — *half size* — function code

The half size statement takes the form .... [hs]

Causes all typeset material produced by the 70|830 VideoComp to be one-half the values specified in all of the control code statements which define the job. The statement must refer to the entire job and must precede text material of any kind; that is, explicit or generated text. This statement applies when point sizes 4.5, 5.5, and 6.5 in size range I are being used. Should any of these sizes be required, all control code parameters which express size must be expressed as twice the point values desired. The **hs** code at the beginning of the controls which specify the job, will reduce all values which are expressed in points to one-half their expressed value.

hy — *hyphenation permitted* — function code

The hyphenation statement takes the form ....[hy{,⟨parameter⟩}$_0^1$]

Instructs the system that hyphenation is requested in the justify horizontally mode. The code is used to reinstate hyphenation if it has been inhibited earlier by the **nh** code or to specify the number of consecutive lines which may be terminated by a hyphen. If the hyphenation permitted function code is used without a parameter, the system assumes that the number of consecutive lines which may be terminated by a hyphen two. The hyphenation process is further defined by the **xd**, exception dictionary, and **nd**, no dictionary, codes. Hyphenation is not provided for the **ce**, **ql**, or **qr** modes.

Note: At system start, [hy,2] is assumed; no more than two consecutive lines will be terminated by a hyphen unless the user specifies otherwise.

ib — *interrupt block* — function code

The interrupt block statement takes the form .... [ib]

Instructs the system that setting in the primary block is to be interrupted in order to set out-of-sequence material in another block. The out-of-sequence material may be, for example, footnotes or a running head. The environment of the primary block will be saved in order that it may be subsequently continued. Environment includes: boundaries, typeface, point size, body lead, justification mode, any tab stops or boundary procedures which are in force, and the vertical and horizontal setting positions at the point of interrupt. If the interrupting block has not yet been started, it must be started and its environment must be specified before the text level mode is re-entered. If the interrupting block has been defined earlier, the code **ib** will cause setting to procede in that block unless a start block code follows.

ig — *ignore* — function code

The ignore statement takes the form .... [ig]

The *ignore* and *consider* function codes provide the capability to include or omit material from a manuscript in order to generate two output versions from a single manuscript. For example, the input:

> .....preceding text [ne,gv1,1[[ig]]]conditional text
> [cn]following text.....

will produce only preceding text and following text if **gv1** is equal to one but will produce preceding text, conditional text, and following text if **gv1** is equal to something other than one. The **cn** cannot, however, be conditional since, if the processor is in ignore mode, the conditional statement will not be executed.

Notes: (1) At system start, the ignore-consider condition is initialized to **cn.** (2) Both text and control following an **ig** code will be ignored until a **cn** code is encountered. (3) If the ignore code appears in any of the four boundary procedures, in a saved string, or in a defined string (for use with the execute defined string code), the consider code must appear before the end of the same procedure or string. (4) The ignore code need not precede every occurrance of the consider code; if the processor is in consider mode, the appearance of a consider code will have no effect.

in — *insert* — alteration operator

The insert statement
takes the form .... [in,⟨line⟩,⟨word space⟩[⟨string⟩]]

Provides the capability to insert material into a composed master file. The *insert* statement is used only after a job has been composed and a master file exists—the alteration is written against the master file. The string in the insert statement carries the material which is to be inserted into the composed master file following the line and word space cited. The first interword space in a line is word space one. The word space must be explicitly stated; zero will not be assumed.

Notes: (1) The **lu** function code establishes the logical unit of the job against which an alteration statement or a series of alteration statements is to apply. (3) The logical unit specified in the start take header is assumed at the beginning of an alteration take. (3) Em, en and thin spaces and space provided by the **hn** function code are not considered to be word spaces for purposes of alteration.

iv — *indirect variable* — global variable

The assign statement
takes the form .... [iv⟨number⟩,⟨parameter⟩]

Provides indirect access to general variables. The value assigned to an *indirect variable* is the number of one of the general variables (an **iv** then may be assigned values from zero through 200 inclusive). The system provides nine indirect variables numbered one through nine. The parameter may be a numeric constant or a variable; but in either case, its value must lie in the zero to 200 range. When an **iv** is used as a parameter, the general variable to which it points is involved in the statement. For example, if **iv1** has the value ten, the statement [ad,iv1,5,4], when executed, will assign the value nine to **gv10**. The only way to specify or change the value assigned to an indirect variable is with the assign statement.

jd — *justify vertically* — function code

The justify vertically statement takes the form .... [jv]

Vertical justification is effected by dividing excess white space—that between the baseline of the last line in the block and the bottom boundary (excluding leading on the last line)—among the justifiable downs in the block. If there are no justifiable down lines in the block, **jd** has no effect. The **jd** code, if present, is effective when the block in which it is in force is terminated.

ju — *justify horizontally* — function code

The justify horizontally statement takes the form .... [ju]

Establishes a typesetting mode in which input material is typeset in lines of equal length; that is, set "justified" between left and right block boundaries. Justification is effected by varying the size of the interword spaces and additionally, perhaps, by hyphenation.

During line composition, word spaces are tentatively assigned the value given in the *basic interword space* typographic variable until a word which would overset the right boundary is encountered. The size of the interword space is then decreased—subject to the minimum value specified in **mi**—in an attempt to fit the entire word on the current line. Failing this, the size of the interword space is increased—subject to the maximum value specified in **mx**—in an attempt to push the entire word to the next line. If this attempt fails and if hyphenation is inhibited, the value of the interword space is increased without limit to push the entire word to the next line.

If hyphenation is permitted and the attempt to push the word to the next line has failed, the word will be hyphenated. Hyphenation, if it is permitted, may be with the exception dictionary to increase accuracy of hyphenation at the expense of speed or without it to increase speed at some loss in accuracy of hyphenation.

Notes: (1) At system start, **ju** is assumed. (2) The justify mode is terminated only by the codes **ce**, **ql**, or **qr**. (3) Em, en, and thin space and fixed space provided by the **hn** code are not altered in horizontal, or line, justification. (4) Partial lines which result from forced line breaks are flushed left. See **xs** for alternative treatment of such partial lines.

jv — *justify vertically* — function code

The justify vertically statement takes the form .... [jv]

Causes text which has been set in a block to be justified vertically by expansion of interlinear space such that the ascenders of the first line in the block are flush against the top boundary and the descenders of the last line are flush against the bottom boundary. The **jv** code is executed when the block in which it is in force is terminated. Surplus space between the last line of text—excluding leading for the last line in the block—and the bottom boundary is allocated to interlinear space in equal increments starting at the top and proceding downward until the surplus space is exhausted. Some lines are not subject to this form of vertical justification; they are: the first line in the text block and any lines initiated by the **dj** or the **dn** function codes. The division of space is in increments of one point for the 70|820 and in increments of 1/32nd point for the 70|830.

lb — *left boundary* — typographic variable

The assign statement takes the form .... [lb,⟨parameter⟩]

Establishes the *left boundary* to be in force and allows direct involvement of *left boundary* in arithmetic operations or conditional procédures. The left boundary is expressed in points rightward from the block origin. A statement which will change the left boundary may be inserted at any point. The boundary change will take effect if or when **cx** is equal to zero; that is, if **cx** is not zero, the change will take effect at the first subsequent natural or forced line break. The parameter may be a numeric constant or a variable. The statement [lb,12] will cause the left boundary to be set twelve points rightward from the block origin, if or when **cx** is equal to zero. The statement [lb,gv57] will cause the left boundary to be set rightward from the block origin (if or when **cx** is equal to zero) by the number of points assigned to **gv57** at the time the statement is executed. Changing the value of **gv57** subsequent to the execution of [lb,gv57] will not in itself change the left boundary .

Notes: (1) The left boundary is initialized to zero at system start. (2) Right boundary must be greater than left boundary.

le — *less than or equal* — conditional operator

The conditional procedure
takes the form .... [le,⟨parameter⟩,⟨parameter⟩[⟨string⟩]]

Provides the capability to execute strings of control codes or to set strings of text or to handle any combination of control codes and text on the condition that the first parameter is *less than or equal* to the second parameter. If the condition is not fulfilled, the string will not be executed. For example, the conditional procedure [le,gv70,gv82[[ad,gv84,gv89,gv70]]] will, if the value of the variable **gv70** is *less than or equal* to the value of the variable **gv82** at the time the procedure is executed, add the values of **gv70** and **gv89** and assign the result to **gv84**. If not, the value of **gv84** will not be altered by the procedure.

ls — *lesser of* — arithmetic operator

The arithmetic statement
takes the form .... [ls,⟨variable⟩,⟨parameter⟩,⟨parameter⟩]

Provides the capability to do a comparison of values and to store the lesser. If the two parameters are equal, their common value is assigned to the variable. The parameters may be numeric constants or they may be variables—the arithmetic operation [ls,gv86,gv85,gv84] will compare the values of **gv85** and **gv84** at the time the operation is executed and assign the value of the lesser to **gv86**.

lt — *less than* — conditional operator

The conditional procedure
takes the form ....[lt,⟨parameter⟩,⟨parameter⟩[⟨string⟩]]

Provides the capability to execute strings of control codes or to set strings of text or to handle any combination of control codes and text on the condition that the first parameter is *less than* the second parameter. If the condition is not fulfilled, the string is not executed. For example, the conditional procedure [lt,gv92,bb[[bb,gv92]]] will, if the value of the variable **gv92** at the time the procedure is executed is *less than* the value assigned to the bottom boundary, assign the value of **gv92** to the bottom boundary. If not, the position of the bottom boundary will not be altered by the procedure.

lu — *logical unit* — alteration operator

The logical unit statement
takes the form .... [lu,⟨parameter⟩]

Used only with alteration takes, the logical unit statement specifies that the alterations to follow—and which appear before a subsequent [lu,⟨parameter⟩]—are to be applied against the logical unit specified by the parameter. The number of the logical unit in the take header is assumed at the start of an alteration take. See manuscript reference point; there are a maximum of 4095 logical units in the PAGE-1 System.

mb — *mark begin* — function code

The mark begin statement takes the form .... [mb,⟨parameter⟩]

The *mark begin* and *mark end* function codes provide the capability to abstract material from the manuscript being set and store it as a string which may be set (or executed) subsequently via the *execute saved string* command. The system has the capacity for two saved strings; strings saved may comprise text only, control only, or text and control. The parameter may be a numeric constant or a variable; but in either case its value must be one or two. The parameter identifies the saved string.

The *mark begin* and *mark end* function codes may be used to save, for example, succeeding first level side heads in the same saved string if the side heads are to be set also as running heads. In this way, when the bottom boundary is encountered, the saved string contains the last side head on the page. Assume that two format statements exist to mark the start and end of a first side head respectively. The first provides space above and changes font and size to that desired for the side head; the second provides space below the side head and changes font and size to that for primary text. *Mark begin* may be added to the end of the first format statement (....;mb,1]) and *mark end* may be added to the beginning of the second ([me;....). The bottom boundary procedure may then include the execute saved string code at the appropriate place to set the contents of saved string number one as the running head (....;es,1;....). The font and size for the running head may, of course, be different—a function of the controls in the bottom boundary procedure which prepared for the running head.

me — *mark end* — function code

The mark end statement takes the form .... [me]

The *mark begin* and *mark end* function codes provide the capability to abstract material from the manuscript being set and store it as a string which may be set (or executed) subsequently via the *execute saved string command.* The **me** command terminates both saved strings; **me** has no parameter.

mi — *minimum interword space* — typographic variable

The assign statement takes the form ....[mi,⟨parameter⟩]

Provides the capability to alter the *minimum interword space* during composition. The minimum interword space is specified in relative units; the relative unit, for VideoFonts supplied by RCA Graphic Systems, is related to the size range of the font. Recommended values for the interword spaces (minimum, basic, and maximum) are specified for and supplied with each digital font. When fonts are loaded on the system disc, these recommended (or perhaps other) interword spaces are loaded also. When a font is specified for use in a job, these stored, "standard" values are used to set the minimum, basic, and maximum interword space typographic variables. Should values different from the "system standard" values be desired in a given job, care must be exercised that the desired values are reinserted each time (after) the font is specified.

ml — *multiply* — arithmetic operator

The arithmetic statement
takes the form .... [ml,⟨variable⟩,⟨parameter⟩,⟨parameter⟩]

Provides the capability to do multiplication. The variable is assigned the numeric (constant) value which accrues when the parameters are multiplied together. The first parameter is the multiplicand; the second is the multiplier. The parameters may be numeric constants or they may be variables—the arithmetic operation [ml,gv96,bl,4] will multiply the value of the typographic variable **bl** at the time the operation is executed by the numeric constant **4** and assign the result to **gv96**.

mx — *maximum interword space* — typographic variable

The assign statement takes the form .... [mx,⟨parameter⟩]

Provides the capability to alter the *maximum interword space* during composition. The maximum interword space is specified in relative units; the relative unit, for VideoFonts supplied by RCA Graphic Systems, is related to the size range of the font. Recommended values for the interword spaces (minimum, basic, and maximum) are specified for and supplied with each digital font. When fonts are loaded on the system disc, these recommended (or perhaps other) interword spaces are loaded also. When a font is specified for use in a job, these stored, "standard" values are used to set the minimum, basic, and maximum interword space typographic variables. Should values different from the "system standard" values be desired in a given job, care must be exercised that the desired values are reinserted each time (after) the font is specified.

nd — *no dictionary* — function code

The no dictionary statement takes the form ....[nd]

Inhibits use of the exception dictionary in the hyphenation process. Inhibition of the exception dictionary results in faster running time with some loss in accuracy of hyphenation.

Note: At system start, use of the exception dictionary is assumed to be permitted.

ne — *not equal* — conditional operator

The conditional procedure
takes the form .... [ne,⟨parameter⟩,⟨parameter⟩[⟨string⟩]]

Provides the capability to execute strings of control codes or to set strings of text or to handle any combination of control codes and text on the condition that two parameters are *not equal.* If the condition is not fulfilled, the string is not executed. For example, the conditional procedure [ne,cy,0[[nl;dn,12]]] will, if the value of **cy** is other than zero, terminate the line being set, provide an additional twelve points of vertical space, and initiate a new line. If not, the procedure will not be executed. This conditional procedure would, in practice be followed by another conditional procedure to direct processor activity if **cy** is *equal* to zero at the time the procedure is executed.

nh — *inhibit hyphenation* — function code

The inhibit hyphenation statement takes the form ....[nh]

Inhibits hyphenation in the horizontal justification process. Each new line will start with the word which encountered the right boundary and caused overset on the preceding line. Interword spaces in the line being justified will be expanded without limit to effect justification. If there are no interword spaces in a line; that is, a word is longer than the distance between left and right boundaries, and hyphenation is inhibited, hyphenation of the word will be forced. Hyphenation, with a limit of two consecutive lines being terminated by hyphens, is assumed at job start.

nl — *new line* — function code

The new line statement takes the form .... [nl{,⟨parameter⟩}$_0^1$]

A forced line break; causes the system to terminate the current line and to initiate a new line starting from the left boundary and spaced downward from the terminated line by a distance equal to the computed lead plus the largest point size encountered in setting the next line. The partial line created by the forced line break will—in the justify mode and if the *expand to fill measure* control is not in force—be set flush left. The new line code may be used with a parameter to specify a number of blank lines before the new line is started. The parameter specified the number of blank lines to be provided between the line terminated and the new line to be started.

Note: See **cy** for the effect of **nl** on the value of **cy**.

nx — *nullify expand to fill measure* — function code

The nullify expand to fill measure statement takes the form .... [nx]

Causes recall from the *expand to fill measure* mode. All partial lines arising from forced line break—in justify mode— will be flushed left. At system start, **nx** is assumed.

ob — *oblique* — function code

The oblique statement takes the form .... [ob]

Causes the VideoComp to introduce an inclination of ten to thirty degrees to the digitized font data in order to generate pseudo-italic characters from font data which were digitized roman or erect. The angle is a factory adjustment nominally about seventeen degrees. In general an *oblique* command is unnecessary with true italic fonts. Roman is assumed at system start.

pd — *page depth* — typographic variable

The assign statement takes the form ....[pd,⟨parameter⟩]

A dedicated general variable which provides a "short hand" means of involving print region depth in arithmetic statements and conditional procedures. The value assigned to **pd** does not fix the depth of the final typeset page; page depth rather, results from the number, relative position, and depth of the blocks which comprise the print region.

Note: At system start, **pd** is initialized to 1584 points (132 picas or about 22 inches).

pg — *new print region* — function code

The new print region statement takes the form .... [pg]

A print region may be viewed as any page produced by the VideoComp. A print region is initiated by the code **pg** which instructs the system to terminate the active (the primary and interrupting if both are defined) blocks on the print region being terminated and to initialize them for the page which is to be started. The code **pg** flushes out the line in process when **cx** is not equal to zero.

Note: Block zero will be saved for the following print region; see discussion under sb.

pi — *standard paragraph indention* — typographic variable

The assign statement takes the form .... [pi,⟨parameter⟩]

Provides the capability to specify the indention to be provided each time that the paragraph function code, **pr**, is executed. The amount of the indention is specified in ens.

Note: At system start, **pi** is initialized at two ens—one em.

pn — *page number counter* — global variable

The assign statement takes the form ....[pn,⟨parameter⟩]

Provides the capability to specify, manipulate, and typeset the page number. The global variable **pn** is actually a dedicated general variable. The page number counter operates on the basis that page numbers are arabic numerals. (Page numbers must be set from the basic subset zero when the **pn** global variable is used.) The page number counter may be initialized to any desired value via the assign statement [pn,⟨parameter⟩]and incremented appropriately with the arithmetic statement [ad,pn,pn,1]. It may then be typeset at the desired time using the set decimal number code [dc,pn]. At system start **pn** is initialized to zero.

pr — *paragraph* — function code

The paragraph statement takes the form .... [pr{,⟨parameter⟩}$_0^1$]

A forced line break; causes the system to terminate the current line and to initiate a new line starting from the left boundary (plus a paragraph indention of the amount specified by the **pi** global variable) and spaced downward from the terminated line by a distance equal to the computed lead plus the largest point size encountered in setting the next line. The partial line created by the forced line break will—in the justify mode and if the *expand to fill measure* control is not in force—be set flush left. The paragraph code may be used with a parameter to specify a number of blank lines before the new paragraph is started. The parameter specifies the number of blank lines to be provided between the line terminated and the new line to be started.

Notes: (1) See **cy** for the effect of **pr** on the value of **cy**. (2) If requirements dictate a movement of the left boundary and a paragraph at the same point in text—for example, extract or secondary text which may be indented and which begins with a paragraph indention—the boundary change should precede the paragraph code in order that the desired results be achieved. (3) The standard paragraph indention is initialized to one em at system start; see **pi**.

ps — *point size* — typographic variable

The assign statement takes the form .... [ps,⟨parameter⟩]

Provides the capability to specify or change the *point size* to be in force and allows direct involvement of point size in arithmetic operations or conditional procedures. The parameter may be a numeric constant or a variable; the statement [ps,10] specifies point size to be ten or the statement [ps,gv11] specifies point size to be the value of **gv11** at the time the statement is executed. Changing the value of **gv11** subsequent to the execution of [ps,gv11] does not in itself change the point size.

Notes: (1) At system start, point size is initialized to eight. (2) Fonts are digitized in several size ranges (two for the 70|820 VideoComp and five for the 70|830) to cover the range of point sizes which the individual machine is capable of setting. Each size range is a separate font, or typeface, and is assigned a unique number when it is stored on the system disc. The point size specified for use must be consistent with the size range of the typeface in force. (3) The same point size may be in two size ranges. When mixing size ranges, consistent resolution demands that consistent point size-typeface combinations be used throughout the job. (4) The PAGE–1 Composition System will provide negative lead capability via point size and body lead combinations; but the net resulting film advance must be positive. (5) Interlinear space or leading is computed by subtracting point size from body lead. The values of **bl** and **ps** used in the computation are the largest values that have been in force in the interval between termination of the last complete line set and the present termination. Computed leading is added to those lines terminated by a natural line break or by the **nl** or **pr** codes. Except that leading is not applied to the last line in a block when the **qc**, **qd**, **jd**, or **jv** commands are in force. (6) The PAGE–1 Composition System will provide a negative lead capability via point size and body lead combinations; but the net resulting film advance must be positive.

pw — *page width* — typographic variable

The assign statement takes the form .... [pw,⟨parameter⟩]

A dedicated general variable which provides a "short hand" means of involving print region width in arithmetic statements and conditional procedures. The value assigned to **pw** does not fix the width of the final typeset page; page width rather, results from the number, relative position, and width of the blocks which comprise the print region.

Note: At system start, **pw** is initialized to 390 points.

qd — *flush down* — function code

The flush down statement takes the form .... [qd]

Causes all lines which have been set in a block to be positioned such that the descenders of the last line are flush to the bottom boundary. The **qd** code is executed when the block in which it is in force is terminated and is then discarded; if it is desired on succeeding pages, it may be put in the bottom boundary procedure. Surplus space between the last line set—excluding leading on the last line—and the bottom boundary is allocated before the first line of text.

ql — *flush left* — function code

The flush left statement takes the form .... [ql]

Establishes a typesetting mode in which text material is set unjustified. The words in a line are separated by the basic interword space; each time a word which would overset the right boundary is encountered, the line is terminated at the preceding interword space and all unused space is inserted at the end of the line. Flush left mode provides a "ragged right" output column. The flush left mode is cancelled by initiating **ce**, **qr**, or **ju** mode.

qo — *quotient* — arithmetic operator

The arithmetic statement
takes the form ....[qo,⟨variable⟩,⟨parameter⟩,⟨parameter⟩]

Provides the capability to do division. The variable is assigned the numeric (constant) value which accrues from the truncated quotient of the two parameters. The first parameter is the dividend; the second is the divisor. The parameters may be numeric constants or they may be variables—the arithmetic operation [qo,gv30,cx,ps] will divide the value of the typographic variable **cx** at the time the operation is executed by the point size in force, truncate the quotient, and assign the result to **gv30**. If, for example, the *point size* in force is eight and the value of **cx** at the time the operation is executed is 161 (points), **gv30** will be assigned the value **20**.

Note: Attempts to divide by zero will result in no action.

qr — *flush right* — function code

The flush right statement takes the form .... [qr]

Establishes a typesetting mode in which text material is set unjustified. The words in a line are separated by the basic interword space; each time a word which would overset the right boundary is encountered, the line is terminated at the preceding interword space and all unused space is inserted at the beginning of the line. Flush right mode provides a "ragged left" output column. The flush right mode is cancelled by initiating **ce**, **ql**, or **ju** mode.

rb — *right boundary* — typographic variable

The assign statement takes the form .... [rb,⟨parameter⟩]

Establishes the *right boundary* to be in force and allows direct involvement of *right boundary* in arithmetic operations or conditional procedures. The right boundary is expressed in points rightward from the block origin. A statement which will change the right boundary may be inserted at any point. The boundary change will take effect if or when **cx** is equal to zero; that is, if **cx** is not zero, the change will take effect at the first subsequent natural or forced line break. The parameter may be a numeric constant or a variable. The statement [rb,288] will cause the right boundary to be set 288 points (24 picas) rightward from the block origin, if or when **cx** is equal to zero. The statement [rb,gv59] will cause the right boundary to be set rightward from block origin (if or when **cx** is equal to zero) by the number of points assigned to **gv59** at the time the statement is executed. Changing the value of **gv59** subsequent to the execution of [rb,gv59] will not in itself change the right boundary .

Notes: (1) The right boundary is initialized to 300 points at system start. (2) Right boundary must be greater than left boundary.

rf — *manuscript reference point* — function code

The manuscript reference point
statement takes the form .... [rf,⟨parameter⟩]

Specifies the beginning of a logical manuscript unit. A job submitted to the PAGE–1 Composition System may comprise one through 4095 logical units; a logical unit is the smallest piece of the job which may be composed or recomposed. Reference points must be assigned in ascending order on the manuscript; they need not necessarily enter the system in that order. The system will arrange each logical unit of the job in ascending order by take number via the manuscript reference point and take header numbers.

rm — *remainder* — arithmetic operator

The arithmetic statement
takes the form .... [rm,⟨variable⟩,⟨parameter⟩,⟨parameter⟩]

Provides the capability to do division. The variable is assigned the numeric (constant) remainder which accrues from the quotient of the two parameters. The first parameter is the dividend; the second is the divisor. The parameters may be numeric constants or they may be variables—the arithmetic operation [rm,gv30,cx,ps] will divide the value of the typographic variable **cx** by the value of the typographic variable **ps** at the time the operation is executed and assign the remainder to **gv30**. If, for example, the *point size* in force is eight and the value of **cx** at the time the operation is executed is 161 (points), **gv30** will be assigned the value **1**.

.....*Continued overleaf*

Note: Attempts to divide by zero will result in no action.

ro — *roman* — function code

The roman statement takes the form .... [ro]

Specifies recall to *roman* after the *oblique*, or pseudoitalic, mode has been in force. Roman is assumed at system start.

rq — *raised quotient* — arithmetic operator

The arithmetic statement
takes the form .... [rq,⟨variable⟩,⟨parameter⟩,⟨parameter⟩]

Provides the capability to do division. The variable is assigned the numeric (constant) value which accrues from the raised quotient of the two parameters. The quotient will be raised if there is a remainder—regardless of its value. The first parameter is the dividend; the second is the divisor. The parameters may be numeric constants or they may be variables—the arithmetic operation [rq,gv31,cx,ps] will divide the value of the typographic variable **cx** by the value of the typographic variable **ps** at the time the operation is executed, raise the quotient if there is a remainder, and assign the result to **gv31**. If, for example, the *point size* in force is eight and the value of **cx** at the time the operation is executed is 161 (points), **gv31** will be assigned the value **21**.

Note: Attempts to divide by zero will result in no action.

rt — *reset tab stop* — function code

The reset tab stops statement takes the form ....[rt]

Provides the capability to cancell all defined tab stops by setting their positions to zero. Tab stops can be cancelled or moved individually only by resetting them individually with the **ts** command.

sb — *start block* — function code

The start block statement takes the form .... [sb,⟨parameter⟩]

Instructs the system that the material which appears in input is to be set in the block specified by the parameter. The parameter may be a numeric constant or a variable and may range in value from zero through 31 inclusive.

PAGE–1 has the capability to handle two blocks concurrently; it is convenient to think of a *primary* block and an *interrupting* block. The code, [ib], causes the processor to store the attributes of the primary block and details regarding the point of text block interruption in order that composition may be resumed when the code [cb] is encountered.

After storing the attributes of the primary block, the processor calls the stored attributes of the interrupting block. The first time that [ib] appears in a job, there is no interrupting block yet defined. In this one instance, the processor assigns the attributes of the primary block to the interrupting block. The user must follow the interrupt block code with a start block code in order to identify the interrupting block; he may also redefine any block attributes which may be different.

When the continue block code is encountered, the process is reversed; the interrupting block attributes are stored and the attributes of the primary block are recalled. For illustration, consider the string of statements and text below. The elements of block definition are omitted for clarity.

[sb,1]One [ib;sb,2]Two [cb]Three [ib]Four
[cb]Five [ib;nl;sb,3]Six [cb]Seven.....

The words One Three Five Seven..... will be set in block one, the words Two Four will be set in block two, and the word Six will be set in block three. At the end of the string, setting will be in process in block one, block two will be closed for the present print region, and block three will be in the interrupted state. Block one is the primary block.

Notes: (1) At system start, setting will begin in block one if no codes to the contrary are encountered. (2) When a new block is started, either primary or interrupting block, it assumes the environment of the block which it replaces as either primary or interrupting block. If any part of the environment of the new block is the same as that of the block it is replacing, that part need not be restated. (3) Block zero has the special characteristic that it—and the material set in it—are saved for the following print region. Provision must be made in the controls to provide for its placement and accommodation.

sn — *snap* — function code

The snap statement takes the form .... [sn]

Specifies that a "snapshot" of the values of the variables and a summary of composition status is to be printed out on the line printer. The capability provides a valuable aid in debugging format statements. In use, the code is explicitly stated at the point in the manuscript at which the snap dump is desired.

st — *start take* — function code

The start take statement
takes the form ...... N[st,⟨job⟩,⟨logical unit⟩,⟨take⟩,b,⟨type⟩,xx]

The header statement used to introduce and identify raw data takes to the system. N is the first character on the input tape; it identifies the keyboard code and a corresponding translation table in the PAGE–1 Composition System. Job, which may have any value in the range one through 999 is the number of the job. Logical unit (1-4095) identifies one of the logical manuscript units into which the job has been divided and take (1-15) identifies one of the keyboard takes into which a logical unit has been divided for ease of keyboarding. The letter **b** specifies the PAGE–1 Composition System. Type specifies the take as original manuscript, **t**, or alteration, **a**. Finally, the letters **xx** provide a two-character identifier, either alphabetic or numeric, for the user's convenience in identifying the keyboard operator, the mark up man, or any other purpose the user may choose.

Notes: (1) The first character in each take must be the keyboard identifier associated with the user's input device. (2) The system has the capability to index and store data associated with 900 jobs; they may be numbered in the range 1 through 999 inclusive. (3) Each input device which generates a different code must be assigned a unique identifier—and the user's PAGE–1 system must incorporate the corresponding translation table.

su — *suppress setting* — function code

The suppress setting
statement takes the form .... [su$\{$,⟨parameter⟩$\}_0^1$]

The *suppress* and *unsuppress* (or restart setting) function codes provide a "trial set" capability. The suppress code causes a flag to be placed in the input string at the point of its occurrence and, in effect, to suppress the recording of "output codes". In suppress mode, all operations which determine horizontal and vertical spacing procede normally.

If the suppress and unsuppress codes are written without parameters or with parameters equal to zero, the material following the flag placed by the suppress code (and extending to the unsuppress code) will be reset when the unsuppress code is encountered. If the suppress code is written with a non-zero parameter, the material set while in suppress mode is saved toward the possibility that it may be acceptable. At the end of the material to be trial set, the unsuppress code may be stated with a non-zero parameter to indicate that the trial set material is to stand or it may be stated without a parameter (or with the parameter equal to zero) to indicate that the trial set material is to be discarded and the material following the flag placed by the suppress code is to be reset.

The suppress setting capability allows computations based upon the material trial set to effect changes in boundaries and other variables before the material is set for the second time. Suppress codes, when setting in the suppress mode, have no effect. No additional flags will be placed in the input string and the unsuppress code, when it appears, will cause a repeat of setting from the earliest suppress code since last in unsuppress mode.

Note: At system start, the suppress-unsuppress mode is reset to **us**.

sw — *set width* — typographic variable

The assign statement takes the form .... [sw,⟨parameter⟩]

Provides the capability to condense or extend the *set width* of the basic font. Each time that *point size* is specified, *set width* is set equal to *point size*; if a set width different from point size is desired, it is necessary to reset **sw** to the desired value each time (after) point size is changed. In general, set width must be equal to a valid point size in the size range of the font being used.

Note: (1) At system start, *set width* is initialized to eight.

sy — *define synonym* — equate operator

The define synonym statement takes the form .... [sy,⟨name⟩[⟨string⟩]]

Provides the capability to equate a two-character, alphanumeric, *name* with a string of text, a string of controls, or a string composed of both text and control. The string may then be specified for execution (or typesetting, if it is a text string) by entering the name into input at the point at which it is to be executed. The first character in the name may be any alphabetic character from **t** through **z** and the second character may be any digit from **1** through **9**—both inclusive. (There are, then, 63 possible synonyms for any job.) Synonyms are job oriented—as opposed to format statements which are global to the system at any installation—and they must be defined in the input for a job prior to the occurrence of their names. Composition terminates when an undefined synonym name appears for execution.

Notes: (1) A synonym name is written as a two-character control word; for example, [z9]. (2) A string given a synonym name may not be recursive; that is, it may not contain the name it is given, nor may it contain another synonym name or format label which contains the name. (3) Strings given synonym names and format statements may contain other synonym names or format labels establishing a chain of such calls. A maximum of forty such calls is permitted in the system. (4) The synonym names **z1** through **z5** may be applied on the keyboard with a single keystroke. This provision simplifies keyboard entry of frequently used synonyms.

tb — *top boundary* — typographic variable

The assign statement takes the form .... [tb,⟨parameter⟩]

Establishes the *top boundary* to be in force and allows direct involvement of *top boundary* in arithmetic operations or conditional procedures. The top boundary is expressed in points below the block origin. A statement which will change the top boundary may be inserted at any point. The boundary change will take effect if or when **cy** is equal to zero. The parameter may be a numeric constant or a variable. The statement [tb,24] will cause the top boundary to be set 24 points (2 picas) below the block origin, if or when **cy** is equal to zero. The statement [tb,gv90] will cause the top boundary to be set below block origin (if or when **cy** is equal to zero) by the number of points assigned to **gv90** at the time the statement is executed. Changing the value of **gv90** subsequent to the execution of [tb,gv90] will not in itself change the top boundary

Notes: (1) The top boundary is initialized to zero at system start. (2) In general, top boundary will be set or changed in the contiguous string of control codes which follows a *start block* statement; if not, care must be exercised in order that the desired results be realized.

tf — *typeface* — typographic variable

The assign statement takes the form .... [tf,⟨parameter⟩]

Provides the means to specify the font in which input material is to be set. In the PAGE–1 Composition System, fonts are digitized and stored on the system disc where they may be retrieved for use as required.

Notes: (1) At system start, the *typeface* typographic variable is initialized to four. Typeface four in the "usual" system delivered by RCA Graphic Systems is Video Times Roman, Range II, Mode C. (2) Fonts are digitized in several size ranges (two for the 70|820 VideoComp and five for the 70|830) to cover the range of point sizes which the individual machine is capable of setting. Each size range is a separate font, or typeface and is assigned a unique number when it is stored on the system disc. The point size specified for use must be consistent with the size range of the typeface in force. (3) The same point size may be in two size ranges. When mixing size ranges, consistent resolution demands that consistent point size-typeface combinations be used throughout the job. (4) The maximum number of characters in a VideoFont is 216.

tn — *test ens for overset* — function code

The test ens for overset statement takes the form .... [tn,⟨parameter⟩]

The *test ens for overset*, when placed at the beginning of a word which is terminated by a control, will test whether that word plus the number of ens specified in the parameter to **tn** will cause overset. If it will, overset is forced. The code is useful when a "word" composed of two components separated only by embedded control is intended to be set as one word. For example, an Arabic numeral footnote citation is generally considered to be part of the word to which it is attached. The citation may, however, be separated from the word by a control and a control embedded in a word is an elegible hyphenation point. In the footnote citation example, it is undesirable for the word to conclude one line and the citation to be first on the next line. The

.....*Continued overleaf*

statement [tn,1] (the parameter is one if the citation is a single digit) must be placed prior to the word which carries the citation. The test ens for overset is executed with the first occurrence of a left delimiter after the code is placed in force; it is then reset automatically.

ts — *set tab stop* — function code

The set tab stop statement
takes the form .... [ts,⟨parameter⟩,⟨parameter⟩]

Causes a tab stop to be set: the first parameter is the number of the tab stop to be set and the second is the position of the tab stop, in points, rightward from the left boundary. The PAGE–1 Composition System has fifteen tab stops numbered one through fifteen.

Notes: (1) Tab stop positions are established with respect to the left boundary; if the left boundary is moved subsequent to setting a tab stop, the tab stop will move with it. (2) A tab stop will not act as a boundary; text can overset tab stops. (3) If the intention is to flush right to or to center between the left boundary and the "first" tab stop on a line, it is necessary to define a tab stop at the left boundary; that is, [ts,1,0]. (4) The right boundary may be used as the final tab stop on a line for purposes of the tab center or the tab right commands. (5) Execute tab statements should be used in justify mode; they may be used in flush left mode. They may not be used in the flush right or center horizontally modes. (6) See **rt** for resetting tab stop positions. (7) At system start, all tab stops are initialized to zero.

up — *baseline up* — function code

The baseline up statement
takes the form .... [up,⟨parameter⟩]

Provides the capability to control the VideoComp baseline. The amount by which the baseline is to be raised from the zero, or nominal, position is specified by the parameter in increments of 1/32nd point. If the output device is the 70|820, the **up** must be specified in increments of one-half point; that is, the parameter must be a multiple of sixteen. Either [dw,0] or [up,0] may be used to restore the baseline to nominal.

Notes: (1) The parameter always specifies an absolute displacement from the nominal baseline position. (2) Current vertical setting position, **cy**, is not affected by an **up** command. (3) At system start, **up** is initialized to zero; that is, baseline position is nominal. (4) The VideoComp aperture limits maximum upward beam displacement for the 70|820 to 38 points and for the 70|830 to 63 and 31/32 points. (5) If a baseline up for the 70|820 is not specified in one-half point increments, the parameter supplied will be rounded up to the nearest multiple of sixteen.

us — *unsuppress setting* — function code

The unsuppress setting statement
takes the form .... [us{,⟨parameter⟩}$_0^1$]

The *unsuppress* code causes the material following the flag placed by an earlier *suppress* code to be reset if the suppress code had no parameter or had a parameter equal to zero. If the unsuppress code has a non-zero parameter (and if the earlier suppress code had a non-zero parameter), the material set during the suppress mode activity is allowed to stand. The suppress and unsuppress (or restart setting) function codes provide a "trial set" capability. The suppress setting capability allows computations based upon the material trial set to effect changes in boundaries and other variables before the material is set for the second time. Unsuppress codes, when setting in the unsuppress mode, have no effect.

Note: At system start, the suppress-unsuppress mode is reset to **us**.

wb — *when bottom* — boundary operator

The boundary procedure takes the form .... [wb[⟨string⟩]]

Provides the capability to retrieve and to execute a string of control codes—or a string composed of control codes and text—whenever the bottom boundary is encountered. If *at bottom* and *when bottom* procedures are both present, the *at bottom* procedure takes precedence. The *at bottom* procedure will be executed at the first bottom boundary encounter; the *when bottom* procedure will be executed at the second and subsequent bottom boundary encounters. If bottom boundary and right boundary procedures are used together, the precedence of execution is established by the order of their first entry into input. For example, consider the following control string fragment: [.....wr[[x1]];wb[[x2]];.....]; the right boundary procedures will take precedence over the bottom boundary procedures throughout the entire job.

Notes: (1) One boundary string must be completed before another is started. For example, if right boundary procedures are to take precedence over bottom boundary procedures when they appear together, care must be exercised to avoid the circumstance in which the bottom boundary is encountered during execution of the right boundary procedure. The use of a **dn**, **dj**, **nl**, or a **pr** within the right boundary procedure *may* result in a bottom boundary encounter—thereby calling the bottom boundary procedure—and composition will terminate. (2) A *when bottom* procedure may be placed in force by simply stating the desired procedure explicitly at any point in the manuscript or it may be included in any of the four boundary procedures. (3) If text is to be set by a *when bottom* procedure, the text must be preceded by a start block command or by the redefinition of the bottom boundary to make the block deeper.

wr — *when right* — boundary operator

The boundary procedure takes the form .... [wr[⟨string⟩]]

Provides the capability to retrieve and to execute a string of control codes—or a string composed of control codes and text—whenever the right boundary is encountered. If *at right* and *when right* procedures are both present, the *at right* procedure takes precedence. The *at right* procedure will be executed at the first right boundary encounter; the *when right* procedure will be executed at the second and subsequent right boundary encounters. If bottom boundary and right boundary procedures are used together, the precedence of execution is established by the order of their first entry into input. For example, in the control string fragment [.....wr[[x1]];wb[[x2]];.....], the right boundary procedures will take precedence over the bottom boundary procedures throughout the entire job.

Notes: (1) One boundary string must be completed before another is started. For example, if right boundary procedures are to take precedence over bottom boundary procedures when they appear together, care must be exercised to avoid the circumstance in which the bottom boundary is encountered during execution of the right boundary procedure. The use of a **dn**, **dj**, **nl**, or a **pr** within the right boundary procedure *may* result in a bottom boundary encounter—thereby calling the bottom boundary procedure—and composition will terminate. (2) A *when right* procedure may be placed in force by simply stating the desired procedure explicitly at any point in the manuscript or it may be included in any of the four boundary procedures. (3) Before execution of the *when right* string, a new line is executed, **cy** is updated, and **cx** is reset to zero.

xc — *tab center* — function code

The execute tab
statement takes the form .... [xc$\{,\langle\text{parameter}\rangle\}_0^1$]

Causes the horizontal setting position to be moved to the tab stop specified by the parameter or to the first tab stop position rightward from the then current horizontal setting position if the tab command has no parameter. The text material which follows—before another execute tab statement or a forced line break—will then be set centered between the tab stop cited, or implied, and:
(1) The tab stop cited by a terminating tab command which has a parameter,
(2) The next larger defined tab stop following **cx** if the terminating tab command has no parameter or if the tab stop cited by the terminating tab command is to the left of **cx**, or
(3) The right boundary if the terminating command is a new line, a paragraph, or a down command.

Any tab stop may be reached from any horizontal setting position. If the cited tab stop is to the left of the current horizontal setting position, the next tab stop to the right of **cx** is used to satisfy a previous tab (or fill to measure if there is one) and then **cx** is set to the cited tab position.

The fill to measure function code may be used with the tab stops; the "fill" takes precedence over the execute tab commands in that excess horizontal space is filled with the fill character(s). The execute tab commands state the limits of the fill to measure action. Up to five *fill to measure* commands with the same or different fill character(s) may be executed between a pair of tab stops; the space to be filled is divided equally among the individual fills.

Notes: (1) Tab stop positions are established with respect to the left boundary; if the left boundary is moved subsequent to setting a tab stop, the tab stop will move with it. (2) A tab stop will not act as a boundary; text can overset tab stops. (3) If the intention is to center between the left boundary and the "first" tab stop on a line, it is necessary to define a tab stop at the left boundary; that is, [ts,1,0]. (4) The right boundary may be used as the final tab stop on a line for purposes of the tab center command.
(5) Execute tab statements should be used in justify mode; they may be used in flush left mode. They may not be used in the flush right or center horizontally modes. (6) See **ts** and **rt** for setting and resetting tab stop positions. (7) At system start, all tab stops are initialized to zero.

xd — *exception dictionary* — function code

The request dictionary statement takes the form .... [xd]

Instructs the system that recourse to the exception dictionary is permitted in the hyphenation process. The **xd** code is assumed at job start; it need not be explicitly stated. The code is used to reinstate use of the exception dictionary if it has been inhibited by an earlier **nd** code.

xr — *tab right* — function code

The execute tab
statement takes the form .... [xr$\{,\langle\text{parameter}\rangle\}_0^1$]

Causes the horizontal setting position to be moved to the tab stop specified by the parameter or to the first tab stop position rightward from the then current horizontal setting position if the tab command has no parameter. The text material which follows—before another execute tab statement or a forced line break—will then be set flush right to:
(1) The tab stop cited by a terminating tab command which has a parameter,
(2) The next larger defined tab stop following **cx** if the terminating tab command has no parameter or if the tab stop cited by the terminating tab command is to the left of **cx**, or
(3) The right boundary if the terminating command is a new line, a paragraph, or a down command.

Any tab stop may be reached from any horizontal setting position. If the cited tab stop is to the left of the current horizontal setting position, the next tab stop to the right of **cx** is used to satisfy a previous tab (or fill to measure if there is one) and then **cx** is set to the cited tab position.

The fill to measure function code may be used with the tab stops; the "fill" takes precedence over the execute tab commands in that excess horizontal space is filled with the fill character(s). The execute tab commands state the limits of the fill to measure action. Up to five fill to measure commands with the same or different fill character(s) may be executed between a pair of tab stops; the space to be filled is divided equally among the individual fills.

Notes: (1) Tab stop positions are established with respect to the left boundary; if the left boundary is moved subsequent to setting a tab stop, the tab stop will move with it. (2) A tab stop will not act as a boundary; text can overset tab stops. (3) If the intention is to flush right to the "first" tab stop on a line, it is necessary to define a tab stop at the left boundary; that is, [ts,1,0]. (4) The right boundary may be used as the final tab stop on a line for purposes of the tab right command. (5) Execute tab statements should be used in justify mode; they may be used in flush left mode. They may not be used in the flush right or center horizontally modes. (6) See **ts** and **rt** for setting and resetting tab stops. (7) At system start, all tab stops are initialized to zero.

xs — *expand to fill measure* — function code

The expand to fill measure
statement takes the form .... [xs$\{,\langle\text{parameter}\rangle\}_0^1$]

Effective only in the horizontal justification mode, the *expand to fill measure* code causes the system to expand the interword spaces in all lines terminated by a forced line break (ie., new line, paragraph, or down) to fill measure. If the parameter is supplied, all lines terminated by a forced line break will be expanded to fill measure if the surplus space is equal to or less than the number of points specified by the parameter; otherwise the partial line will be flushed left. If the parameter is omitted, all lines terminated by a forced line break will be expanded without limit to fill measure.

Note: This mode of operation is cancelled by the code **nx**.

xt — *tab left* — function code

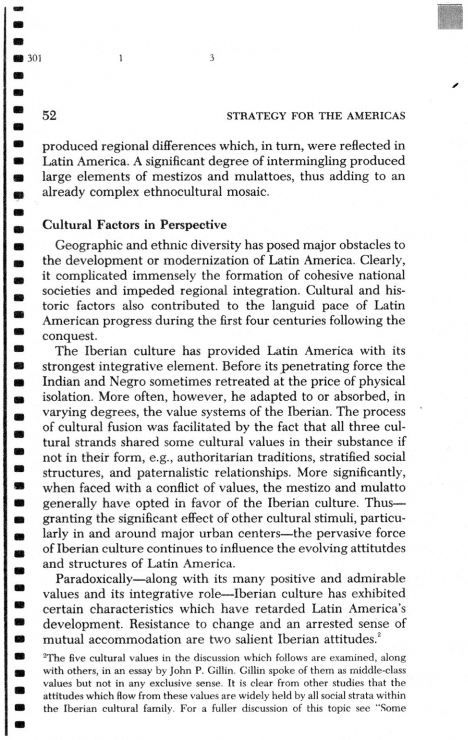
The execute tab
statement takes the form .... [xt{,⟨parameter⟩}$_0^1$]

Causes the horizontal setting position to be moved to the tab stop specified by the parameter or to the first tab stop position rightward from the then current horizontal setting position if the tab command has no parameter. The text material which follows—before another execute tab statement or a forced line break—will then set flush left to that tab stop. Any tab stop may be reached from any horizontal setting position. If the cited tab stop is to the left of the current horizontal setting position, the next tab stop to the right of **cx** is used to satisfy a previous tab (or fill to measure if there is one) and then **cx** is set to the cited tab position. The fill to measure function code may be used with the tab stops; the "fill" takes precedence over the execute tab commands in that excess horizontal space is filled with the fill character(s). The execute tab commands state the limits of the fill to measure action. Up to five fill to measure commands with the same or different fill character(s) may be executed between a pair of tab stops; the space to be filled is divided equally among the individual fills.

Notes: (1) Tab stop positions are established with respect to the left boundary; if the left boundary is moved subsequent to setting a tab stop, the tab stop will move with it. (2) A tab stop will not act as a boundary; text can overset tab stops. (3) Execute tab statements should be used in justify mode; they may be used in flush left mode. They may not be used in the flush right or center horizontally modes. (4) See **ts** and **rt** for setting and resetting tab stop positions. (5) At system start, all tab stops are initialized to zero.

A PAGE-1 format manual

This format manual comprises generalized—and in some cases refined—versions of the format statements developed and implied in chapters five through eight.



301 1 3

52 STRATEGY FOR THE AMERICAS

produced regional differences which, in turn, were reflected in Latin America. A significant degree of intermingling produced large elements of mestizos and mulattoes, thus adding to an already complex ethnocultural mosaic.

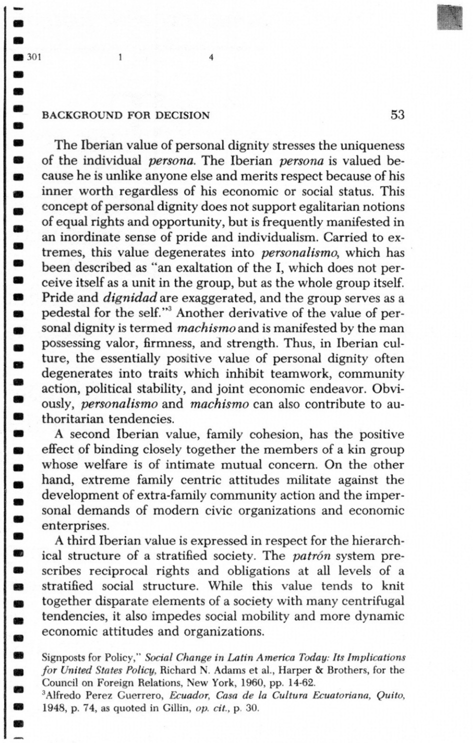
Cultural Factors in Perspective

Geographic and ethnic diversity has posed major obstacles to the development or modernization of Latin America. Clearly, it complicated immensely the formation of cohesive national societies and impeded regional integration. Cultural and historic factors also contributed to the languid pace of Latin American progress during the first four centuries following the conquest.

The Iberian culture has provided Latin America with its strongest integrative element. Before its penetrating force the Indian and Negro sometimes retreated at the price of physical isolation. More often, however, he adapted to or absorbed, in varying degrees, the value systems of the Iberian. The process of cultural fusion was facilitated by the fact that all three cultural strands shared some cultural values in their substance if not in their form, e.g., authoritarian traditions, stratified social structures, and paternalistic relationships. More significantly, when faced with a conflict of values, the mestizo and mulatto generally have opted in favor of the Iberian culture. Thus—granting the significant effect of other cultural stimuli, particularly in and around major urban centers—the pervasive force of Iberian culture continues to influence the evolving attitutdes and structures of Latin America.

Paradoxically—along with its many positive and admirable values and its integrative role—Iberian culture has exhibited certain characteristics which have retarded Latin America's development. Resistance to change and an arrested sense of mutual accommodation are two salient Iberian attitudes.[2]

[2] The five cultural values in the discussion which follows are examined, along with others, in an essay by John P. Gillin. Gillin spoke of them as middle-class values but not in any exclusive sense. It is clear from other studies that the attitudes which flow from these values are widely held by all social strata within the Iberian cultural family. For a fuller discussion of this topic see "Some



301 1 4

BACKGROUND FOR DECISION 53

The Iberian value of personal dignity stresses the uniqueness of the individual *persona.* The Iberian *persona* is valued because he is unlike anyone else and merits respect because of his inner worth regardless of his economic or social status. This concept of personal dignity does not support egalitarian notions of equal rights and opportunity, but is frequently manifested in an inordinate sense of pride and individualism. Carried to extremes, this value degenerates into *personalismo,* which has been described as "an exaltation of the I, which does not perceive itself as a unit in the group, but as the whole group itself. Pride and *dignidad* are exaggerated, and the group serves as a pedestal for the self."[3] Another derivative of the value of personal dignity is termed *machismo* and is manifested by the man possessing valor, firmness, and strength. Thus, in Iberian culture, the essentially positive value of personal dignity often degenerates into traits which inhibit teamwork, community action, political stability, and joint economic endeavor. Obviously, *personalismo* and *machismo* can also contribute to authoritarian tendencies.

A second Iberian value, family cohesion, has the positive effect of binding closely together the members of a kin group whose welfare is of intimate mutual concern. On the other hand, extreme family centric attitudes militate against the development of extra-family community action and the impersonal demands of modern civic organizations and economic enterprises.

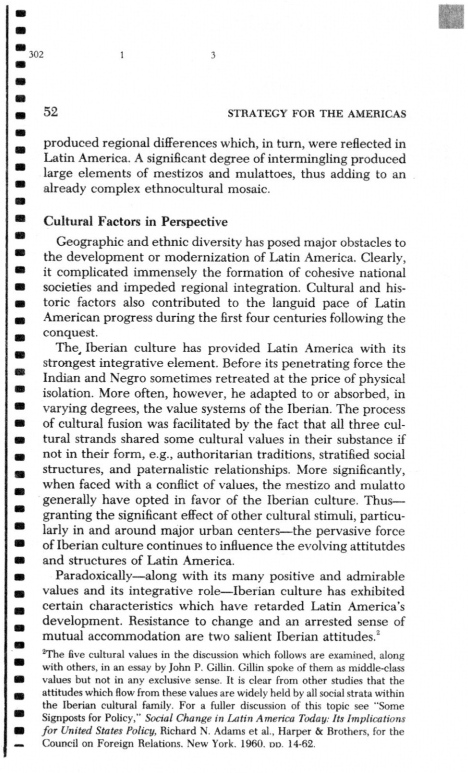
A third Iberian value is expressed in respect for the hierarchical structure of a stratified society. The *patrón* system prescribes reciprocal rights and obligations at all levels of a stratified social structure. While this value tends to knit together disparate elements of a society with many centrifugal tendencies, it also impedes social mobility and more dynamic economic attitudes and organizations.

Signposts for Policy," *Social Change in Latin America Today: Its Implications for United States Policy,* Richard N. Adams et al., Harper & Brothers, for the Council on Foreign Relations, New York, 1960, pp. 14-62.

[3] Alfredo Perez Guerrero, *Ecuador, Casa de la Cultura Ecuatoriana, Quito,* 1948, p. 74, as quoted in Gillin, *op. cit.,* p. 30.

Using [F3] and [F4], the carryover portion is placed at the bottom of the following page above any footnotes cited on that page.

Using [F3] and [F5], the carryover portion is allowed to hang below nominal page depth.



302 1 3

52 STRATEGY FOR THE AMERICAS

produced regional differences which, in turn, were reflected in Latin America. A significant degree of intermingling produced large elements of mestizos and mulattoes, thus adding to an already complex ethnocultural mosaic.

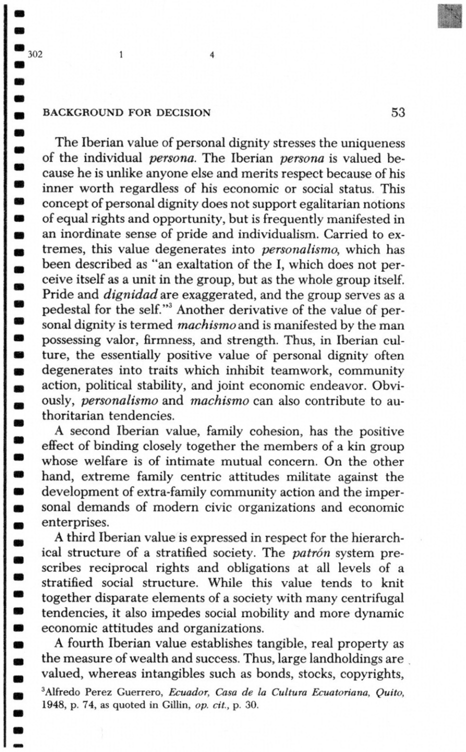
Cultural Factors in Perspective

Geographic and ethnic diversity has posed major obstacles to the development or modernization of Latin America. Clearly, it complicated immensely the formation of cohesive national societies and impeded regional integration. Cultural and historic factors also contributed to the languid pace of Latin American progress during the first four centuries following the conquest.

The Iberian culture has provided Latin America with its strongest integrative element. Before its penetrating force the Indian and Negro sometimes retreated at the price of physical isolation. More often, however, he adapted to or absorbed, in varying degrees, the value systems of the Iberian. The process of cultural fusion was facilitated by the fact that all three cultural strands shared some cultural values in their substance if not in their form, e.g., authoritarian traditions, stratified social structures, and paternalistic relationships. More significantly, when faced with a conflict of values, the mestizo and mulatto generally have opted in favor of the Iberian culture. Thus—granting the significant effect of other cultural stimuli, particularly in and around major urban centers—the pervasive force of Iberian culture continues to influence the evolving attitutdes and structures of Latin America.

Paradoxically—along with its many positive and admirable values and its integrative role—Iberian culture has exhibited certain characteristics which have retarded Latin America's development. Resistance to change and an arrested sense of mutual accommodation are two salient Iberian attitudes.[2]

[2] The five cultural values in the discussion which follows are examined, along with others, in an essay by John P. Gillin. Gillin spoke of them as middle-class values but not in any exclusive sense. It is clear from other studies that the attitudes which flow from these values are widely held by all social strata within the Iberian cultural family. For a fuller discussion of this topic see "Some Signposts for Policy," *Social Change in Latin America Today: Its Implications for United States Policy,* Richard N. Adams et al., Harper & Brothers, for the Council on Foreign Relations, New York, 1960, pp. 14-62.



302 1 4

BACKGROUND FOR DECISION 53

The Iberian value of personal dignity stresses the uniqueness of the individual *persona.* The Iberian *persona* is valued because he is unlike anyone else and merits respect because of his inner worth regardless of his economic or social status. This concept of personal dignity does not support egalitarian notions of equal rights and opportunity, but is frequently manifested in an inordinate sense of pride and individualism. Carried to extremes, this value degenerates into *personalismo,* which has been described as "an exaltation of the I, which does not perceive itself as a unit in the group, but as the whole group itself. Pride and *dignidad* are exaggerated, and the group serves as a pedestal for the self."[3] Another derivative of the value of personal dignity is termed *machismo* and is manifested by the man possessing valor, firmness, and strength. Thus, in Iberian culture, the essentially positive value of personal dignity often degenerates into traits which inhibit teamwork, community action, political stability, and joint economic endeavor. Obviously, *personalismo* and *machismo* can also contribute to authoritarian tendencies.

A second Iberian value, family cohesion, has the positive effect of binding closely together the members of a kin group whose welfare is of intimate mutual concern. On the other hand, extreme family centric attitudes militate against the development of extra-family community action and the impersonal demands of modern civic organizations and economic enterprises.

A third Iberian value is expressed in respect for the hierarchical structure of a stratified society. The *patrón* system prescribes reciprocal rights and obligations at all levels of a stratified social structure. While this value tends to knit together disparate elements of a society with many centrifugal tendencies, it also impedes social mobility and more dynamic economic attitudes and organizations.

A fourth Iberian value establishes tangible, real property as the measure of wealth and success. Thus, large landholdings are valued, whereas intangibles such as bonds, stocks, copyrights,

[3] Alfredo Perez Guerrero, *Ecuador, Casa de la Cultura Ecuatoriana, Quito,* 1948, p. 74, as quoted in Gillin, *op. cit.,* p. 30.

Sample pages from
Strategy for the Americas
by J. W. Reidy
Copyright © 1966 by McGraw-Hill, Inc.
Used with the Publisher's permission.

```
C1 ≡ [by,1,0;by,2,gv121;pd,gv87;gv89,gv87;
      gv194,0;gv182,0;ad,pn,pn,1;fn,1;
      sb,1;lb,0;tb,0;rb,gv59;bb,gv87;ju;ib;
      sb,2;lb,0;tb,0;rb,gv59;bb,gv87;ps,gv15;bl,ps;tf,gv45;Y4nl;
      sb,3;bl,gv29;ju;wb[[Y3]];cb;
      ab[[gv89,gv88;pd,gv186;Y1;by,1,gv123;by,2,0]];wb[[Y1]]]

C2 ≡ [jd;pg;sb,1;bb,gv89;gv194,0;
      eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                  ad,gv194,gv193,gv173;ad,gv90,gv193,gv172;
                  df,bb,gv89,gv90;gv182,0]];
      ad,pn,pn,1;ib;sb,2;bb,gv89;Y2;nl;sb,3;ju;cb]

C3 ≡ [gv192,cy;sb,0;bb,gv89;gv182,1]

C4 ≡ [jd;pg;sb,1;bb,gv89;ad,pn,pn,1;ib;
      sb,2;bb,gv89;Y2;nl;sb,3;ju;cb]

C5 ≡ [gv192,cy;bb,gv89;gv182,1]

[sy,Y1[[C2]];sy,Y2[[**]];sy,Y3[[C3]];sy,Y4[[††]]]
[sy,Y1[[C4]];sy,Y2[[**]];sy,Y3[[C5]];sy,Y4[[††]]]
```

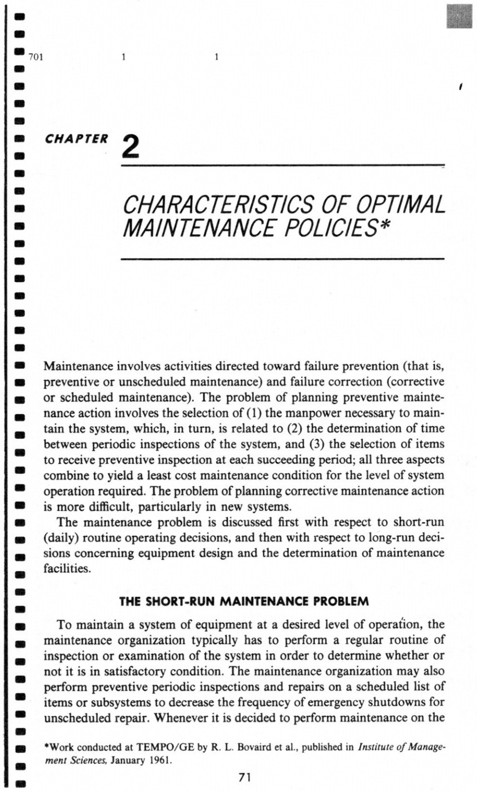
** Desired running head. †† Desired drop folio.

```
F3 ≡ [ad,gv90,cy,ps;ad,gv90,gv90,gv172;ad,gv90,gv90,gv194;
      ib;df,bb,gv89,gv90;ps,gv25;tf,gv55;up,gv174;dc,fn;up,0;
      ps,gv14;tf,gv44]

F4 ≡ [nl;eq,gv182,0[[df,gv192,cy,gv173]];
        eq,gv182,1[[df,gv193,cy,gv173]];cb;
      df,gv160,pd,gv192;by,3,gv160;ad,gv90,gv192,gv172;
      ne,gv194,0[[ad,gv90,gv90,gv194;
                  df,gv161,gv160,gv194;by,0,gv161]];
      df,bb,gv89,gv90;df,gv175,ps,gv25;ml,gv175,gv175,32;
      gv2,ps;gv4,tf;ps,gv25;tf,gv55;up,gv175;dc,fn;up,0;
      ps,gv2;tf,gv4;ad,fn,fn,1]

F5 ≡ [nl;eq,gv182,0[[df,gv192,cy,gv173]];gv182,0;cb;
      df,gv160,pd,gv192;by,3,gv160;ad,gv90,gv192,gv172;
      df,bb,gv89,gv90;df,gv175,ps,gv25;ml,gv175,gv175,32;
      gv2,ps;gv4,tf;ps,gv25;tf,gv55;up,gv175;dc,fn;up,0;ps,gv2;
      tf,gv4;ad,fn,fn,1]
```

THIS SET OF FORMAT STATEMENTS will generate single-column pages with footnotes. The chapter opening page has a drop folio which may be specified to be centered or flush outside by defining the synonym Y4 to be G6 or G3 respectively. Continuation pages provide for a user specified running head which is selected by definition of the synonym Y2.

The footnote statements provide for footnotes to be placed at the bottom of the page on which they are cited; two options are provided to handle footnotes which will not fit entirely on their citation page. If C2 and C3 are selected as text block and footnote block bottom boundary procedures, the carryover portion will be placed at the bottom of the following page above any footnotes which are cited on that page. The format statements to start and close footnotes if this option is elected are F3 and F4. If C4 and C5 are selected as text block and footnote block bottom boundary procedures, the carryover portion will hang below nominal page depth. In this case F3 and F5 are used to start and close footnotes.

Footnote citation in either option is by sequential, superscript, arabic numerals created by use of smaller point size and raised baseline.



701 1 1

CHAPTER 2

CHARACTERISTICS OF OPTIMAL MAINTENANCE POLICIES*

Maintenance involves activities directed toward failure prevention (that is, preventive or unscheduled maintenance) and failure correction (corrective or scheduled maintenance). The problem of planning preventive maintenance action involves the selection of (1) the manpower necessary to maintain the system, which, in turn, is related to (2) the determination of time between periodic inspections of the system, and (3) the selection of items to receive preventive inspection at each succeeding period; all three aspects combine to yield a least cost maintenance condition for the level of system operation required. The problem of planning corrective maintenance action is more difficult, particularly in new systems.

The maintenance problem is discussed first with respect to short-run (daily) routine operating decisions, and then with respect to long-run decisions concerning equipment design and the determination of maintenance facilities.

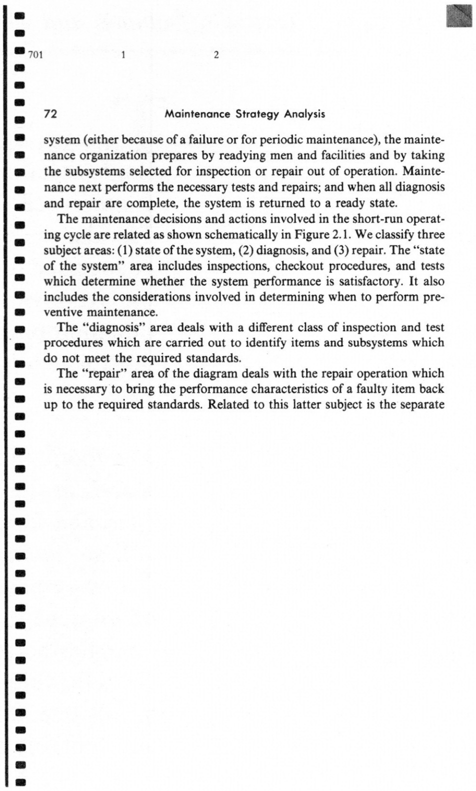
THE SHORT-RUN MAINTENANCE PROBLEM

To maintain a system of equipment at a desired level of operation, the maintenance organization typically has to perform a regular routine of inspection or examination of the system in order to determine whether or not it is in satisfactory condition. The maintenance organization may also perform preventive periodic inspections and repairs on a scheduled list of items or subsystems to decrease the frequency of emergency shutdowns for unscheduled repair. Whenever it is decided to perform maintenance on the system (either because of a failure or for periodic maintenance), the maintenance organization prepares by readying men and facilities and by taking the subsystems selected for inspection or repair out of operation. Maintenance next performs the necessary tests and repairs; and when all diagnosis and repair are complete, the system is returned to a ready state.

---

*Work conducted at TEMPO/GE by R. L. Bovaird et al., published in *Institute of Management Sciences*, January 1961.

71

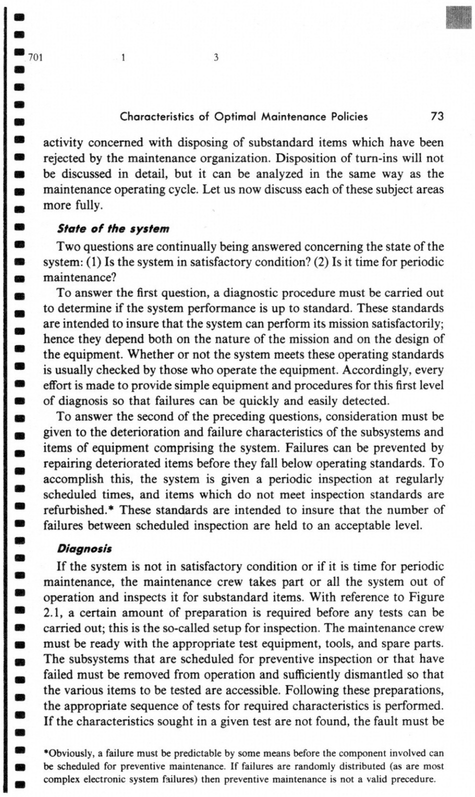


701 1 2

The maintenance decisions and actions involved in the short-run operating cycle are related as shown schematically in Figure 2.1. We classify three subject areas: (1) state of the system, (2) diagnosis, and (3) repair. The "state of the system" area includes inspections, checkout procedures, and tests which determine whether the system performance is satisfactory. It also includes the considerations involved in determining when to perform preventive maintenance.

The "diagnosis" area deals with a different class of inspection and test procedures which are carried out to identify items and subsystems which do not meet the required standards.

The "repair" area of the diagram deals with the repair operation which is necessary to bring the performance characteristics of a faulty item back up to the required standards. Related to this latter subject is the separate activity concerned with disposing of substandard items which have been rejected by the maintenance organization. Disposition of turn-ins will not be discussed in detail, but it can be analyzed in the same way as the maintenance operating cycle. Let us now discuss each of these subject areas more fully.



701 1 3

State of the system

Two questions are continually being answered concerning the state of the system: (1) Is the system in satisfactory condition? (2) Is it time for periodic maintenance?

To answer the first question, a diagnostic procedure must be carried out to determine if the system performance is up to standard. These standards are intended to insure that the system can perform its mission satisfactorily; hence they depend both on the nature of the mission and on the design of the equipment. Whether or not the system meets these operating standards is usually checked by those who operate the equipment. Accordingly, every effort is made to provide simple equipment and procedures for this first level of diagnosis so that failures can be quickly and easily detected.

To answer the second of the preceding questions, consideration must be given to the deterioration and failure characteristics of the subsystems and items of equipment comprising the system. Failures can be prevented by repairing deteriorated items before they fall below operating standards. To accomplish this, the system is given a periodic inspection at regularly scheduled times, and items which do not meet inspection standards are refurbished.* These standards are intended to insure that the number of failures between scheduled inspection are held to an acceptable level.

Diagnosis

If the system is not in satisfactory condition or if it is time for periodic maintenance, the maintenance crew takes part or all the system out of operation and inspects it for substandard items. With reference to Figure 2.1, a certain amount of preparation is required before any tests can be carried out; this is the so-called setup for inspection. The maintenance crew must be ready with the appropriate test equipment, tools, and spare parts. The subsystems that are scheduled for preventive inspection or that have failed must be removed from operation and sufficiently dismantled so that the various items to be tested are accessible. Following these preparations, the appropriate sequence of tests for required characteristics is performed. If the characteristics sought in a given test are not found, the fault must be

---

*Obviously, a failure must be predictable by some means before the component involved can be scheduled for preventive maintenance. If failures are randomly distributed (as are most complex electronic system failures) then preventive maintenance is not a valid precedure.

The white space citation in this illustration appears in line seven of the typeset page 72 as follows: "...in Figure 2.1.[gv6,276;c9] We classify..."; since the requested space is greater than one-half column and appears early enough on the citation page, space is allocated at the bottom of the page.

Sample pages from
MAINTAINABILITY: A Major Element of System Effectiveness
by A. S. Goldman and T. B. Slattery
Copyright © 1964 by John Wiley and Sons, Inc.
Used with the Publisher's permission.

```
C6 ≡ [by,1,0;by,2,gv121;pd,gv87;gv89,gv87;gv86,gv88;
     gv194,0;gv182,0;gv195,0;gv143,gv123;gv144,gv123;
     ad,pn,pn,1;fn,1;
     sb,1;lb,0;tb,0;rb,gv59;bb,gv87;ju;ib;
     sb,2;lb,0;tb,0;rb,gv59;bb,gv87;ps,gv15;bl,ps;tf,gv45;Y4;nl;
     sb,3;ps,gv14;bl,gv29;tf,gv44;ju;wb[[Y3]];cb;
     ab[[gv89,gv88;pd,gv186;Y1;by,2,0]];wb[[Y1]]]

C7 ≡ [eq,gv160,0[[jd]];pg;
     sb,1;by,1,gv143;bb,gv86;gv194,0;gv144,gv123;
     eq,gv182,1[[df,gv161,pd,gv193;by,0,gv161;
                 ad,gv194,gv193,gv173;ad,gv90,gv193,gv172;
                 df,bb,bb,gv90;gv182,0]];
     eq,gv195,1[[gv144,gv143;gv195,0]];
     ad,pn,pn,1;fn,1;ib;sb,2;bb,gv86;Y2;nl;
     sb,3;ps,gv14;tf,gv44;ju;cb;
     gv86,gv88;gv143,gv123;gv160,0]

C8 ≡ [gv192,cy;sb,0;bb,gv86;gv182,1]

[sy,Y1[[C7]];sy,Y2[[**]];sy,Y3[[C8]];sy,Y4[[††]]]
```

** Desired running head. †† Desired drop folio.

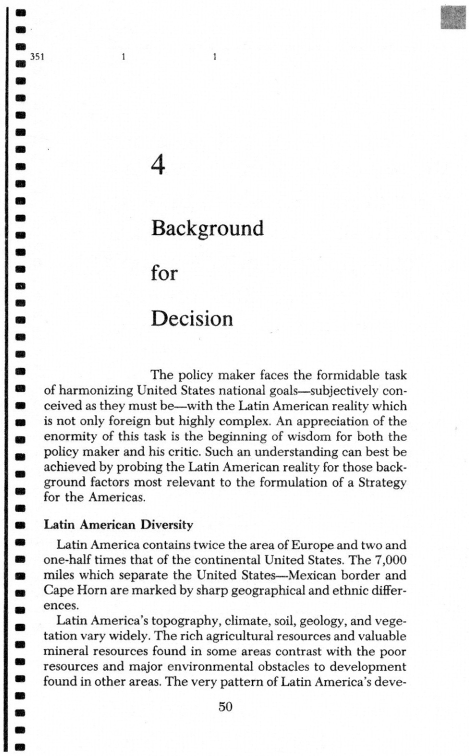
```
F6 ≡ [ad,gv90,cy,ps;ad,gv90,gv90,gv172;
     ad,gv90,gv90,gv194;ib;df,bb,gv89,gv90;
     lt,gv89,gv90[[df,bb,gv90,gv89]];
     eq,fn,1[*];eq,fn,2[†];eq,fn,3[‡];
     eq,fn,4[§];eq,fn,5[¶]]

F7 ≡ [nl;eq,gv182,0[[df,gv192,cy,gv173]];
       eq,gv182,1[[df,gv193,cy,gv173]];cb;
     df,gv160,pd,gv192;by,3,gv160;ad,gv90,gv192,gv172;
     ne,gv194,0[[ad,gv90,gv90,gv194;
                 df,gv161,gv160,gv194;by,0,gv161]];
     df,bb,gv89,gv90;
     eq,fn,1[*];eq,fn,2[†];eq,fn,3[‡];
     eq,fn,4[§];eq,fn,5[¶[fn,0]];ad,fn,fn,1]
```

THIS SET OF FORMAT STATEMENTS will generate single-column pages with footnotes and allocate white space for subsequent "strip in" of graphics. The chapter opening page has a drop folio which may be specified to be centered or flush outside by defining the synonym Y4 to be G6 or G3 respectively. Continuation pages provide for a user specified running head which is selected by definition of the synonym Y2.

The footnote statements provide for footnotes to be placed at the bottom of the page on which they are cited and for carryover of footnote fragments which will not fit standard page depth to the bottom of the following page above any footnotes which are cited on that page. The format statements to start and close footnotes are F6 and F7. Footnote citation is by the following sequence of characters: *, †, ‡, §, and ¶, full size and set on the same baseline as primary and footnote text.

White space is cited by assigning to gv6 a value equal to the depth of white space desired (in points) and then calling C9; the amount of white space requested should equal an integral number of text lines.

```
C9 ≡ [eq,gv195,1[[ge,gv86,gv6[[ne,gv143,gv123[[qo,gv91,gv86,2;lt,gv6,gv91[[ad,gv143,gv143,gv6]];df,gv86,gv86,gv6]]]]]];

     eq,gv195,0[[eq,cy,0[[eq,cx,0[[gv90,0]];ne,cx,0[[gv90,ps]]]];
                 ne,cy,0[[eq,cx,0[[df,gv90,cy,bl;ad,gv90,gv90,ps]];ne,cx,0[[ad,gv90,cy,ps]]]];df,gv90,bb,gv90;
                 ge,gv90,gv6[[qo,gv91,bb,2;lt,gv6,gv91[[ad,gv144,gv144,gv6;by,1,gv144]];df,bb,bb,gv6]];
                 lt,gv90,gv6[[lt,gv6,gv92[[ad,gv143,gv123,gv6]];df,gv86,gv86,gv6;gv195,1]]]]]
```



351 1 1

4

Background

for

Decision

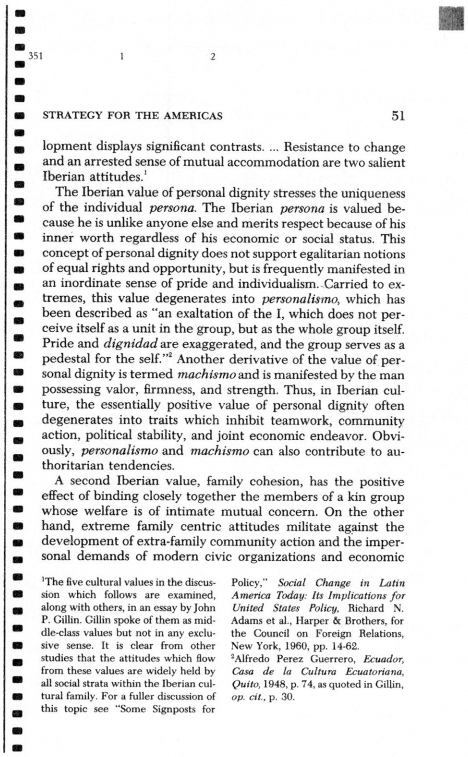
The policy maker faces the formidable task of harmonizing United States national goals—subjectively conceived as they must be—with the Latin American reality which is not only foreign but highly complex. An appreciation of the enormity of this task is the beginning of wisdom for both the policy maker and his critic. Such an understanding can best be achieved by probing the Latin American reality for those background factors most relevant to the formulation of a Strategy for the Americas.

Latin American Diversity

Latin America contains twice the area of Europe and two and one-half times that of the continental United States. The 7,000 miles which separate the United States—Mexican border and Cape Horn are marked by sharp geographical and ethnic differences.

Latin America's topography, climate, soil, geology, and vegetation vary widely. The rich agricultural resources and valuable mineral resources found in some areas contrast with the poor resources and major environmental obstacles to development found in other areas. The very pattern of Latin America's deve-

50



351 1 2

STRATEGY FOR THE AMERICAS 51

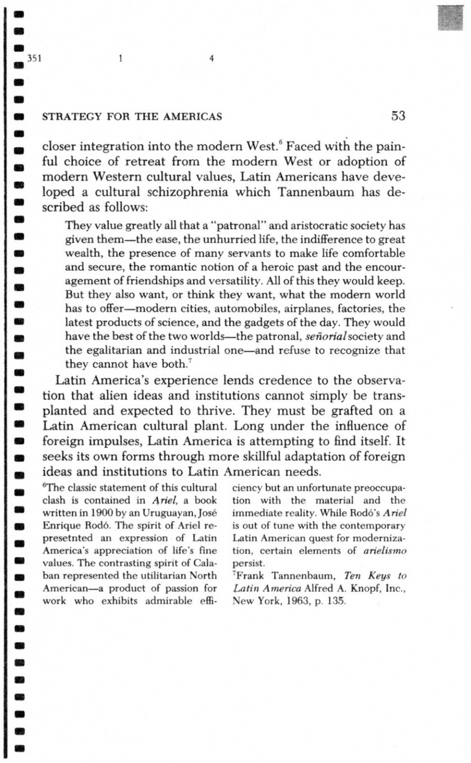
lopment displays significant contrasts. ... Resistance to change and an arrested sense of mutual accommodation are two salient Iberian attitudes.[1]

The Iberian value of personal dignity stresses the uniqueness of the individual *persona.* The Iberian *persona* is valued because he is unlike anyone else and merits respect because of his inner worth regardless of his economic or social status. This concept of personal dignity does not support egalitarian notions of equal rights and opportunity, but is frequently manifested in an inordinate sense of pride and individualism. Carried to extremes, this value degenerates into *personalismo,* which has been described as "an exaltation of the I, which does not perceive itself as a unit in the group, but as the whole group itself. Pride and *dignidad* are exaggerated, and the group serves as a pedestal for the self."[2] Another derivative of the value of personal dignity is termed *machismo* and is manifested by the man possessing valor, firmness, and strength. Thus, in Iberian culture, the essentially positive value of personal dignity often degenerates into traits which inhibit teamwork, community action, political stability, and joint economic endeavor. Obviously, *personalismo* and *machismo* can also contribute to authoritarian tendencies.

A second Iberian value, family cohesion, has the positive effect of binding closely together the members of a kin group whose welfare is of intimate mutual concern. On the other hand, extreme family centric attitudes militate against the development of extra-family community action and the impersonal demands of modern civic organizations and economic

[1] The five cultural values in the discussion which follows are examined, along with others, in an essay by John P. Gillin. Gillin spoke of them as middle-class values but not in any exclusive sense. It is clear from other studies that the attitudes which flow from these values are widely held by all social strata within the Iberian cultural family. For a fuller discussion of this topic see "Some Signposts for Policy," *Social Change in Latin America Today: Its Implications for United States Policy,* Richard N. Adams et al., Harper & Brothers, for the Council on Foreign Relations, New York, 1960, pp. 14-62.

[2] Alfredo Perez Guerrero, *Ecuador, Casa de la Cultura Ecuatoriana, Quito,* 1948, p. 74, as quoted in Gillin, *op. cit.,* p. 30.



351 1 4

STRATEGY FOR THE AMERICAS 53

closer integration into the modern West.[6] Faced with the painful choice of retreat from the modern West or adoption of modern Western cultural values, Latin Americans have developed a cultural schizophrenia which Tannenbaum has described as follows:

> They value greatly all that a "patronal" and aristocratic society has given them—the ease, the unhurried life, the indifference to great wealth, the presence of many servants to make life comfortable and secure, the romantic notion of a heroic past and the encouragement of friendships and versatility. All of this they would keep. But they also want, or think they want, what the modern world has to offer—modern cities, automobiles, airplanes, factories, the latest products of science, and the gadgets of the day. They would have the best of the two worlds—the patronal, *señorial* society and the egalitarian and industrial one—and refuse to recognize that they cannot have both.[7]

Latin America's experience lends credence to the observation that alien ideas and institutions cannot simply be transplanted and expected to thrive. They must be grafted on a Latin American cultural plant. Long under the influence of foreign impulses, Latin America is attempting to find itself. It seeks its own forms through more skillful adaptation of foreign ideas and institutions to Latin American needs.

[6] The classic statement of this cultural clash is contained in *Ariel,* a book written in 1900 by an Uruguayan, José Enrique Rodó. The spirit of Ariel represetnted an expression of Latin America's appreciation of life's fine values. The contrasting spirit of Calaban represented the utilitarian North American—a product of passion for work who exhibits admirable efficiency but an unfortunate preoccupation with the material and the immediate reality. While Rodó's *Ariel* is out of tune with the contemporary Latin American quest for modernization, certain elements of *arielismo* persist.

[7] Frank Tannenbaum, *Ten Keys to Latin America* Alfred A. Knopf, Inc., New York, 1963, p. 135.

Page 53 simulates the final page of a chapter. It is shorter than the other sample pages because the footnotes are placed below the last line of chapter text by a fixed space.

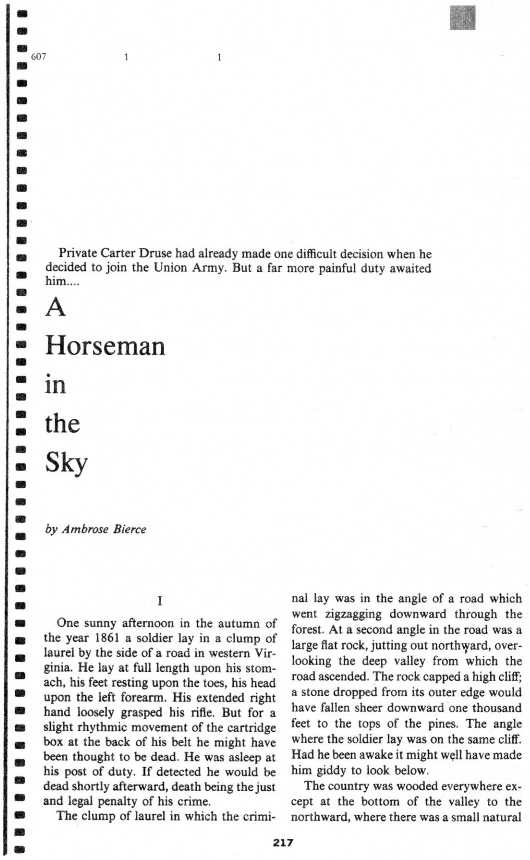
Sample pages from
Strategy for the Americas
by J. W. Reidy
Copyright © 1966 by McGraw-Hill, Inc.
Used with the Publisher's permission.

```
L1 ≡ [by,1,0;by,2,gv121;pd,gv87;gv89,gv87;ml,gv191,gv87,2;
      gv192,gv87;ad,pn,pn,1;fn,1;gv190,0;gv183,1;su;
      sb,1;lb,0;tb,0;rb,gv59;bb,gv192;ju;ib;
      sb,2;lb,0;tb,0;rb,gv59;bb,gv88;ps,gv15;bl,gv29;tf,gv45;Y4;nl;
      sb,3;lb,0;rb,gv159;bb,gv191;ju;wr[[ad,gv190,gv190,1]];
      wb[[eq,gv183,1[[gv192,gv189;gv191,gv193;gv183,0;us]];
          eq,gv183,0[[sb,5;bb,gv88]]]];cb;
      ab[[eq,gv183,1[[gv183,0;us]];
          eq,gv183,0[[df,gv160,pd,gv191;by,3,gv160;by,5,gv160;
                      pd,gv186;gv89,gv88;ml,gv191,gv88,2;
                      gv192,gv88;ad,pn,pn,1;gv190,0;pg;
                      by,1,gv123;by,2,0;gv183,1;su;
                      sb,1;lb,0;tb,0;rb,gv59;bb,gv192;ju;ib;
                      sb,2;lb,0;tb,0;rb,gv59;bb,gv88;Y2;nl;
                      sb,3;lb,0;rb,gv159;bb,gv191;ju;
                      wr[[ad,gv190,gv190,1]];
                      wb[[eq,gv183,1[[gv192,gv189;
                                      gv191,gv193;gv183,0;us]];
                          eq,gv183,0[[sb,5;bb,gv88]]]];cb]]]];
      wb[[eq,gv183,1[[gv183,0;us]];
          eq,gv183,0[[df,gv160,pd,gv191;by,3,gv160;by,5,gv160;
                      ml,gv191,gv88,2;gv192,gv88;ad,pn,pn,1;
                      gv190,0;pg;gv183,1;su;
                      sb,1;lb,0;tb,0;rb,gv59;bb,gv192;ju;ib;
                      sb,2;lb,0;tb,0;rb,gv59;bb,gv88;Y2;nl;
                      sb,3;lb,0;rb,gv159;bb,gv191;ju;
                      wr[[ad,gv190,gv190,1]];
                      wb[[eq,gv183,1[[gv192,gv189;
                                      gv191,gv193;gv183,0;us]];
                          eq,gv183,0[[sb,5;bb,gv88]]]];cb]]]]]

L2 ≡ [eq,gv183,1[[gv189,cy;ad,gv90,cy,ps;ad,gv91,gv90,gv172;
                df,gv91,gv89,gv91;ib;ml,bb,gv91,2;ps,gv25;
                tf,gv55;up,gv174;dc,fn;up,0;ps,gv14;tf,gv44]];
      eq,gv183,0[[ib;ps,gv25;tf,gv55;up,gv174;dc,fn;up,0;
                ps,gv14;tf,gv44]]]

L3 ≡ [eq,gv183,1[[ne,cx,0[[ad,gv190,gv190,1]];rq,gv191,gv190,2;
                ml,gv191,gv191,gv29;df,gv191,gv191,gv173;
                df,gv192,gv89,gv191;df,gv192,gv192,gv172;
                nl;cb;bb,gv192;df,gv175,ps,gv25;
                ml,gv175,gv175,32;gv2,ps;gv4,tf;ps,gv25;
                tf,gv55;up,gv175;dc,fn;up,0;ps,gv2;tf,gv4;
                ad,fn,fn,1;ar[[gv193,gv191]]]]];
      eq,gv183,0[[nl;cb;df,gv175,ps,gv25;ml,gv175,gv175,32;
                gv2,ps;gv4,tf;ps,gv25;tf,gv55;up,gv175;
                dc,fn;up,0;ps,gv2;tf,gv4;ad,fn,fn,1]]]

L5 ≡ [eq,gv183,1[[ad,gv189,cy,ps;df,gv189,gv192,gv189;
                ad,gv189,gv189,gv191;gv183,0;us]];
      eq,gv183,0[[df,gv160,pd,gv189;by,3,gv160;by,5,gv160]]]
```

THIS SET OF FORMAT STATEMENTS will generate single-column pages with two-column footnotes. The chapter opening page has a drop folio which may be specified to be centered or flush outside by defining the synonym Y4 to be G6 or G3 respectively. Continuation pages provide for a user specified running head which is selected by definition of the synonym Y2.

This footnote treatment is common in those books that have a very high density of short footnotes. The footnote statements to start and close footnotes are L2 and L3; they provide for footnotes to be placed at the bottom of the page on which they are cited and in two columns of equal depth. If the total number of footnote lines is odd, the odd line will be assigned to the left-hand footnote column. Footnotes do not carry from one page to the next; if a footnote will not fit, the text line which cites the footnote, and the footnote, are pushed forward to the following page. On the final page of the chapter, the footnote columns are placed below the last line of text by a user specified amount of space. The label L5 must follow the last text character of each chapter; that is, it concludes the data for the chapter.

The style generator I8 will compute and assign appropriate values to the general variables. Follow the *Strategy for the Americas* example in Chapter 10; note however, that text block and footnote block bottom boundary procedures are written into L1 and need not be specified in the job specification.



607 1 1

Private Carter Druse had already made one difficult decision when he decided to join the Union Army. But a far more painful duty awaited him....

A Horseman in the Sky

by Ambrose Bierce

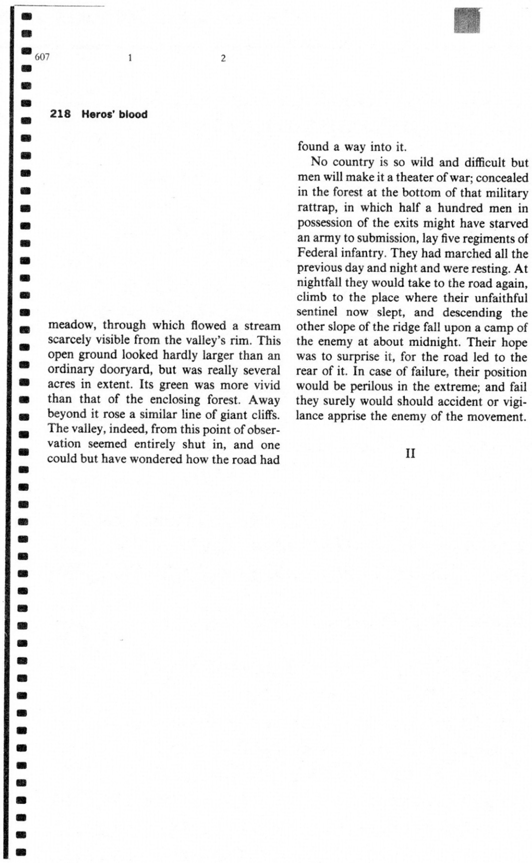
I

One sunny afternoon in the autumn of the year 1861 a soldier lay in a clump of laurel by the side of a road in western Virginia. He lay at full length upon his stomach, his feet resting upon the toes, his head upon the left forearm. His extended right hand loosely grasped his rifle. But for a slight rhythmic movement of the cartridge box at the back of his belt he might have been thought to be dead. He was asleep at his post of duty. If detected he would be dead shortly afterward, death being the just and legal penalty of his crime.

The clump of laurel in which the criminal lay was in the angle of a road which went zigzagging downward through the forest. At a second angle in the road was a large flat rock, jutting out northward, overlooking the deep valley from which the road ascended. The rock capped a high cliff; a stone dropped from its outer edge would have fallen sheer downward one thousand feet to the tops of the pines. The angle where the soldier lay was on the same cliff. Had he been awake it might well have made him giddy to look below.

The country was wooded everywhere except at the bottom of the valley to the northward, where there was a small natural

217



607 1 2

218 Heros' blood

meadow, through which flowed a stream scarcely visible from the valley's rim. This open ground looked hardly larger than an ordinary dooryard, but was really several acres in extent. Its green was more vivid than that of the enclosing forest. Away beyond it rose a similar line of giant cliffs. The valley, indeed, from this point of observation seemed entirely shut in, and one could but have wondered how the road had found a way into it.

No country is so wild and difficult but men will make it a theater of war; concealed in the forest at the bottom of that military rattrap, in which half a hundred men in possession of the exits might have starved an army to submission, lay five regiments of Federal infantry. They had marched all the previous day and night and were resting. At nightfall they would take to the road again, climb to the place where their unfaithful sentinel now slept, and descending the other slope of the ridge fall upon a camp of the enemy at about midnight. Their hope was to surprise it, for the road led to the rear of it. In case of failure, their position would be perilous in the extreme; and fail they surely would should accident or vigilance apprise the enemy of the movement.

II

General variable 123 has the value one for this sample—the result is the running head format.

The white space at the top and bottom of column one of page 218 is called by the citation [gv6,276;B8;gv6,598;B8;pr] at the end of line twelve of column two on page 217. The white space at the bottom of column two of page 218 is called by the citation [gv6,596;B8;pr] at the end of the last line of text in column one page 218.

Sample pages from
ACCENT: U.S.A.
Copyright © 1965 by Scott, Foresman and Company
"A Horseman in the Sky" by Ambrose Bierce
Used with the Publisher's permission.

```
B3 ≡ [gv143,gv123;gv149,gv123;gv168,gv88;gv169,gv88;
     gv195,0;gv196,0;gv197,0;gv198,0;gv200,0;iv1,98;
     gv1,0;gv3,3;ad,pn,pn,1;fn,1;
     sb,6;lb,0;tb,0;rb,gv58;bb,gv88;pd,gv88;
     ne,gv123,0[[bb,gv87;pd,gv87]];ib;
     sb,2;lb,0;rb,gv58;bb,gv88;ps,gv15;bl,ps;tf,gv45;Y4;nl;
     sb,3;lb,0;rb,gv59;bl,gv29;ju;wb[[Y3]];cb;
     ab[[jv;sb,4;bb,gv99;eq,gv182,0[[ib;sb,5;cb]];gv182,0;
         iv1,99;gv1,1;gv3,5;ab[[jv;pg;sb,1;bb,gv168;ad,pn,pn,1;
         pd,gv88;ne,gv123,0[[pd,gv186;by,2,0]];ib;sb,2;lb,0;rb,gv58;
         Y2;nl;sb,3;lb,0;rb,gv59;ju;cb;by,1,gv143;by,4,gv149;
         gv98,gv168;gv99,gv169;gv144,gv143;gv150,gv149;
         gv195,gv197;gv196,gv198;gv168,gv88;gv169,gv88;
         gv143,gv123;gv149,gv123;gv197,0;gv198,0;
         gv200,0;iv1,98;gv1,0;gv3,3]]]];wb[[Y1]]]

B4 ≡ [ad,gv144,cy,gv110;gv150,gv144;nl;by,1,gv144;by,4,gv150;
     df,gv98,gv88,gv144;ne,gv123,0[[df,gv98,gv87,gv144]];
     gv99,gv98;sb,1;lb,0;rb,gv59;bb,gv98;ps,gv11;bl,gv26;tf,gv41;ju]

B6 ≡ [gv192,cy;bb,gv88;gv182,1;eq,gv1,0[[sb,5]]]

B7 ≡ [eq,gv1,0[[jd;sb,4;bb,gv99;eq,gv182,0[[ib;sb,5;cb]];
               gv182,0;iv1,99;gv1,2;gv3,5]];
     eq,gv1,1[[jd;pg;sb,1;bb,gv168;ad,pn,pn,1;ib;
               sb,2;lb,0;rb,gv58;Y2;nl;sb,3;lb,0;rb,gv59;
               ju;cb;by,1,gv143;by,4,gv149;gv98,gv168;
               gv99,gv169;gv144,gv143;gv150,gv149;gv195,gv197;
               gv196,gv198;gv168,gv88;gv169,gv88;gv143,gv123;
               gv149,gv123;gv197,0;gv198,0;gv200,0;iv1,98;gv3.3]];
     df,gv1,gv1,1]

[sy,Y1[[B7]];sy,Y2[[**]];sy,Y3[[B6]];sy,Y4[[‡‡]]]
```

** Desired running head. †† Desired drop folio.

```
lt,gv90,gv6[[ge,gv6,gv92[[gv197,1]];lt,gv6,gv92[[ad,gv143,gv143,gv6]];df,gv168,gv168,
```

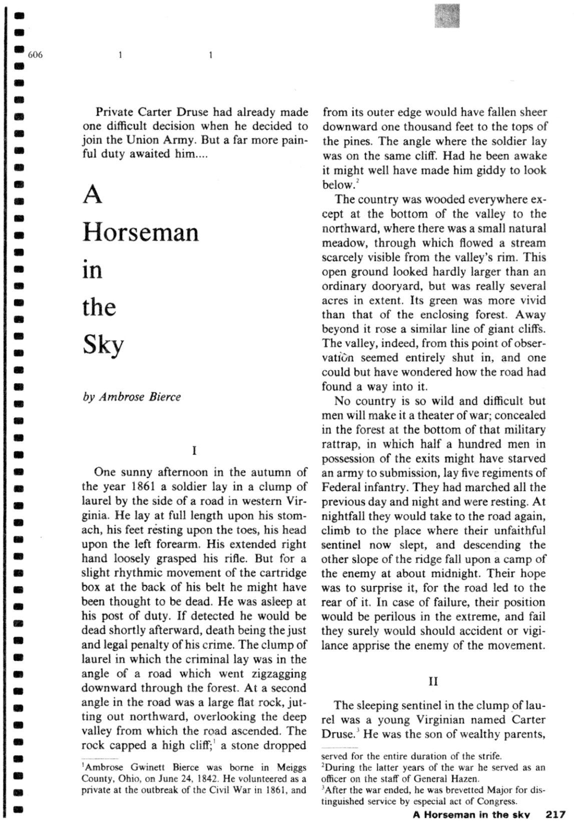
THIS SET OF FORMAT STATEMENTS will generate two-column pages with footnotes and allocate white space for subsequent "strip in" of graphics. The format statements B3 and B4 are used together to provide for a full, two-column chapter opening followed by two columns of text to complete the chapter opening page. The format statements B7 and B6 are the text block and footnote block bottom boundary procedures respectively and they must be equated with the synonym names Y1 and Y3 in the job specification.

White space is cited by assigning to gv6 a value equal to the depth of white space desired (in points) and then calling B8; the amount of white space requested should equal an integral number of text lines.

The footnote format statements appear and their use is described on page B11.

The two-column style generator (for use with these format statements) provides for user selection of two basic typographic formats. See discussion on page B11.

```
B8 ≡ [eq,gv200,1[[eq,gv198,0[[eq,gv197,1[[ge,gv169,gv6[[ge,gv6,gv92[[gv198,1]];
                                           lt,gv6,gv92[[ne,gv149,gv123[[gv198,1]];eq,gv149,gv123[[ad,gv149,gv149,gv6]]]];
                                           df,gv169,gv169,gv6]]]];
                          eq,gv197,0[[lt,gv168,gv6[[ge,gv169,gv6[[ge,gv6,gv92[[gv198,1]];lt,gv6,gv92[[ne,gv149,gv123[[gv198,1]];
                                                                                                 eq,gv149,gv123[[ad,gv149,gv149,gv6]]]];
                                                         df,gv169,gv169,gv6]]]];
                                 ge,gv168,gv6[[ge,gv6,gv92[[gv197,1]];
                                           lt,gv6,gv92[[ne,gv143,gv123[[gv197,1]];eq,gv143,gv123[[ad,gv143,gv143,gv6]]]];
                                           df,gv168,gv168,gv6]]]]]]]];

     eq,gv200,0[[eq,gv196,1[[ge,gv6,gv92[[gv197,1]];lt,gv6,gv92[[ad,gv143,gv143,gv6]];df,gv168,gv168,gv6;gv200,1]];
              eq,gv196,0[[eq,cy,0[[gv90,0;ne,cx,0[[gv90,ps]]]];
                          ne,cy,0[[eq,cx,0[[df,gv90,cy,bl;ad,gv90,gv90,ps]];ne,cx,0[[ad,gv90,cy,ps]]]];df,gv90,bb,gv90;
                          eq,gv1,0[[eq,gv195,1[[ge,gv6,gv92[[gv196,1]];
                                          lt,gv6,gv92[[ne,gv150,gv123[[gv196,1]];eq,gv150,gv123[[ad,gv150,gv150,gv6;by,4,gv150]]]];
                                          df,gv99,gv99,gv6]];
                                eq,gv195,0[[ge,gv90,gv6[[ge,gv6,gv92[[gv195,1]];
                                                      lt,gv6,gv92[[ne,gv144,gv123[[gv195,1]];
                                                                   eq,gv144,gv123[[ad,gv144,gv144,gv6;by,1,gv144]]]];
                                                      df,bb,bb,gv6;gv98,bb]];
                                          lt,gv90,gv6[[ge,gv6,gv92[[gv196,1]];lt,gv6,gv92[[ad,gv150,gv150,gv6;by,4,gv150]];
                                          df,gv99,gv99,gv6]]]]]];
                          eq,gv1,1[[ge,gv90,gv6[[ge,gv6,gv92[[gv196,1]];lt,gv6,gv92[[ne,gv150,gv123[[gv196,1]];
                                                                                          eq,gv150,gv123[[ad,gv150,gv150,gv6;by,4,gv150]]]];
                                          df,bb,bb,gv6;gv99,bb]];
```



Private Carter Druse had already made one difficult decision when he decided to join the Union Army. But a far more painful duty awaited him....

A Horseman in the Sky

by Ambrose Bierce

I

One sunny afternoon in the autumn of the year 1861 a soldier lay in a clump of laurel by the side of a road in western Virginia. He lay at full length upon his stomach, his feet resting upon the toes, his head upon the left forearm. His extended right hand loosely grasped his rifle. But for a slight rhythmic movement of the cartridge box at the back of his belt he might have been thought to be dead. He was asleep at his post of duty. If detected he would be dead shortly afterward, death being the just and legal penalty of his crime. The clump of laurel in which the criminal lay was in the angle of a road which went zigzagging downward through the forest. At a second angle in the road was a large flat rock, jutting out northward, overlooking the deep valley from which the road ascended. The rock capped a high cliff;[1] a stone dropped from its outer edge would have fallen sheer downward one thousand feet to the tops of the pines. The angle where the soldier lay was on the same cliff. Had he been awake it might well have made him giddy to look below.[2]

The country was wooded everywhere except at the bottom of the valley to the northward, where there was a small natural meadow, through which flowed a stream scarcely visible from the valley's rim. This open ground looked hardly larger than an ordinary dooryard, but was really several acres in extent. Its green was more vivid than that of the enclosing forest. Away beyond it rose a similar line of giant cliffs. The valley, indeed, from this point of observation seemed entirely shut in, and one could but have wondered how the road had found a way into it.

No country is so wild and difficult but men will make it a theater of war; concealed in the forest at the bottom of that military rattrap, in which half a hundred men in possession of the exits might have starved an army to submission, lay five regiments of Federal infantry. They had marched all the previous day and night and were resting. At nightfall they would take to the road again, climb to the place where their unfaithful sentinel now slept, and descending the other slope of the ridge fall upon a camp of the enemy at about midnight. Their hope was to surprise it, for the road led to the rear of it. In case of failure, their position would be perilous in the extreme, and fail they surely would should accident or vigilance apprise the enemy of the movement.

II

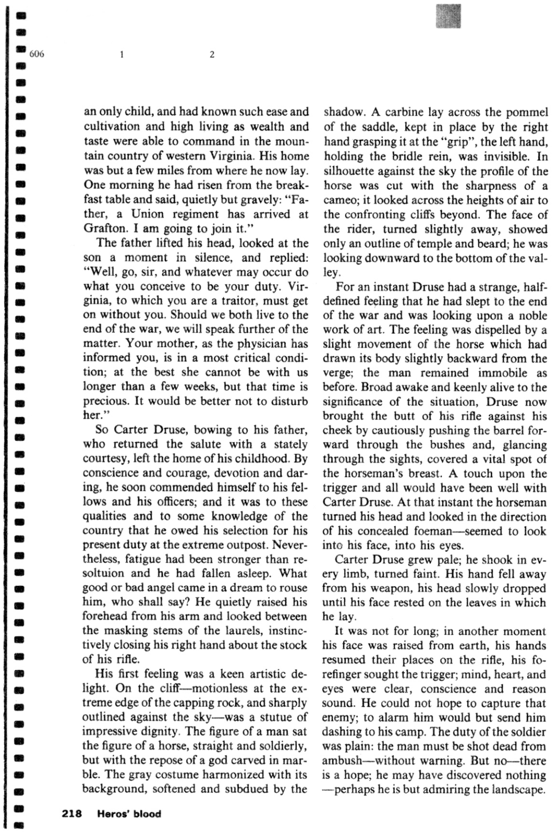
The sleeping sentinel in the clump of laurel was a young Virginian named Carter Druse.[3] He was the son of wealthy parents,

---

[1] Ambrose Gwinett Bierce was borne in Meiggs County, Ohio, on June 24, 1842. He volunteered as a private at the outbreak of the Civil War in 1861, and served for the entire duration of the strife.

[2] During the latter years of the war he served as an officer on the staff of General Hazen.

[3] After the war ended, he was brevetted Major for distinguished service by especial act of Congress.



an only child, and had known such ease and cultivation and high living as wealth and taste were able to command in the mountain country of western Virginia. His home was but a few miles from where he now lay. One morning he had risen from the breakfast table and said, quietly but gravely: "Father, a Union regiment has arrived at Grafton. I am going to join it."

The father lifted his head, looked at the son a moment in silence, and replied: "Well, go, sir, and whatever may occur do what you conceive to be your duty. Virginia, to which you are a traitor, must get on without you. Should we both live to the end of the war, we will speak further of the matter. Your mother, as the physician has informed you, is in a most critical condition; at the best she cannot be with us longer than a few weeks, but that time is precious. It would be better not to disturb her."

So Carter Druse, bowing to his father, who returned the salute with a stately courtesy, left the home of his childhood. By conscience and courage, devotion and daring, he soon commended himself to his fellows and his officers; and it was to these qualities and to some knowledge of the country that he owed his selection for his present duty at the extreme outpost. Nevertheless, fatigue had been stronger than resoltuion and he had fallen asleep. What good or bad angel came in a dream to rouse him, who shall say? He quietly raised his forehead from his arm and looked between the masking stems of the laurels, instinctively closing his right hand about the stock of his rifle.

His first feeling was a keen artistic delight. On the cliff—motionless at the extreme edge of the capping rock, and sharply outlined against the sky—was a stutue of impressive dignity. The figure of a man sat the figure of a horse, straight and soldierly, but with the repose of a god carved in marble. The gray costume harmonized with its background, softened and subdued by the shadow. A carbine lay across the pommel of the saddle, kept in place by the right hand grasping it at the "grip", the left hand, holding the bridle rein, was invisible. In silhouette against the sky the profile of the horse was cut with the sharpness of a cameo; it looked across the heights of air to the confronting cliffs beyond. The face of the rider, turned slightly away, showed only an outline of temple and beard; he was looking downward to the bottom of the valley.

For an instant Druse had a strange, half-defined feeling that he had slept to the end of the war and was looking upon a noble work of art. The feeling was dispelled by a slight movement of the horse which had drawn its body slightly backward from the verge; the man remained immobile as before. Broad awake and keenly alive to the significance of the situation, Druse now brought the butt of his rifle against his cheek by cautiously pushing the barrel forward through the bushes and, glancing through the sights, covered a vital spot of the horseman's breast. A touch upon the trigger and all would have been well with Carter Druse. At that instant the horseman turned his head and looked in the direction of his concealed foeman—seemed to look into his face, into his eyes.

Carter Druse grew pale; he shook in every limb, turned faint. His hand fell away from his weapon, his head slowly dropped until his face rested on the leaves in which he lay.

It was not for long; in another moment his face was raised from earth, his hands resumed their places on the rifle, his forefinger sought the trigger; mind, heart, and eyes were clear, conscience and reason sound. He could not hope to capture that enemy; to alarm him would but send him dashing to his camp. The duty of the soldier was plain: the man must be shot dead from ambush—without warning. But no—there is a hope; he may have discovered nothing —perhaps he is but admiring the landscape.

General variable 123 has the value zero for this sample—the result is the running foot format.

Note that the footnote in column one will not fit completely in column one; it is carried over to column two and placed above the footnotes cited in that column. The footnote which will not fit entirely in column two is allowed to hang below nominal page depth.

Sample pages from
ACCENT: U.S.A.
Copyright © 1965 by Scott, Foresman and Company
"A Horseman in the Sky" by Ambrose Bierce
Used with the publisher's permission.

```
B5 ≡ [by,1,0;by,4,0;gv144,0;gv150,0;gv143,gv123;gv149,gv123;
     gv98,gv88;pd,gv88;ne,gv123,0[[gv98,gv87;pd,gv87]];
     gv99,gv98;gv168,gv88;gv169,gv88;gv195,0;gv196,0;
     gv197,0;gv198,0;gv200,0;iv1,98;gv1,0;gv3,3;ad,pn,pn,1;fn,1;
     sb,1;lb,0;tb,0;rb,gv59;bb,gv98;ib;
     sb,2;lb,0;rb,gv58;bb,gv88;ps,gv15;bl,gv29;tf,gv45;Y4;nl;
     sb,3;lb,0;rb,gv59;ju;wb[[Y3]];cb;wb[[Y1]]]

B6 ≡ [gv192,cy;bb,gv88;gv182,1;eq,gv1,0[[sb,5]]]

B7 ≡ [eq,gv1,0[[jd;sb,4;bb,gv99;eq,gv182,0[[ib;sb,5;cb]];
             gv182,0;iv1,99;gv1,2;gv3,5]];
     eq,gv1,1[[jd;pg;sb,1;bb,gv168;ad,pn,pn,1;ib;
             sb,2;lb,0;rb,gv58;Y2;nl;sb,3;lb,0;rb,gv59;
             ju;cb;by,1,gv143;by,4,gv149;gv98,gv168;
             gv99,gv169;gv144,gv143;gv150,gv149;gv195,gv197;
             gv196,gv198;gv168,gv88;gv169,gv88;gv143,gv123;
             gv149,gv123;gv197,0;gv198,0;gv200,0;iv1,98;gv3,3]];
     df,gv1,gv1,1]

[sy,Y1[[B7]];sy,Y2[[**]];sy,Y3[[B6]];sy,Y4[[‡‡]]]
```

** Desired running head. †† Desired drop folio.

```
F8 ≡ [ad,gv90,cy,ps;ad,gv90,gv90,gv172;ib;
     df,bb,iv1,gv90;ps,gv25;tf,gv55;up,gv174;
     dc,fn;up,0;ps,gv14;tf,gv44]

F9 ≡ [nl;eq,gv182,0[[df,gv192,cy,gv173]];
       eq,gv182,1[[df,gv193,cy,gv173]];
                  eq,gv1,0[[df,gv161,pd,gv193;by,5,gv161;
                            ad,gv91,gv193,gv172]]]];cb;
     df,gv160,pd,gv192;by,gv3,gv160;ad,gv90,gv192,gv172;
     df,bb,iv1,gv90;df,gv175,ps,gv25;ml,gv175,gv175,32;
     gv2,ps;gv4,tf;ps,gv25;tf,gv55;up,gv175;dc,fn;up,0;
     ps,gv2;tf,gv4;ad,fn,fn,1]
```

THIS SET OF FORMAT STATEMENTS will generate two-column pages with footnotes and allocate white space for subsequent "strip in" of graphics. The format statement B5 provides a chapter opening page with two columns; it is assumed that the chapter title occupies the left hand column. The format statements B7 and B6 are the text block and footnote block bottom boundary procedures respectively; they must be equated with the synonym names Y1 and Y3 in the job specification.

The footnote format statements are F8 and F9 to open and close footnotes respectively. The footnote treatment provides for carryover from column one to column two (above any footnotes cited in column two); column two footnotes which will not fit are allowed to hang below nominal page depth.

The white space capability and its use are described on page B9.

The two-column style generator (for use with these format statements) provides for user selection of two basic typographic formats. If gv123 is set to zero in the job specification, a folio and running foot will be provided. If this option is chosen, Y2 and Y4 should both be given the same running head definition. If gv123 is set to one in the job specification, the chapter opening page will have a drop folio and continuation pages will have running heads. In this latter case, the drop folio may be centered of flushed outside by defining Y2 to be G6 or G3 respectively.

```
18 ≡ [df,gv88,gv60,1;ml,gv88,gv88,gv26;ad,gv88,gv88,gv11;
     qo,gv92,gv88,2;
     gr,gv188,gv15,gv16;ad,gv123,gv188,gv95;ad,gv186,gv123,gv88;
     df,gv121,gv186,gv15;df,gv87,gv121,gv94;ml,pi,gv76,2;
     df,gv159,gv59,gv57;qo,gv159,gv159,2;
     ad,gv128,gv122,gv59;ad,gv128,gv128,gv57;
     df,gv173,gv29,gv14;df,gv174,gv14,gv25;ml,gv174,gv174,32;
     ad,gv102,gv102,gv11;df,gv103,gv102,gv26;ml,gv61,gv61,gv12;
     ad,gv104,gv104,gv12;df,gv105,gv104,gv27;
     ad,gv111,gv111,gv11;df,gv112,gv111,gv26;ad,gv113,gv113,gv20;
     eq,gv77,1[[gv77,gv76;ml,gv137,gv77,gv20]];
     ad,gv114,gv114,gv11;df,gv115,gv114,gv26;ad,gv116,gv116,gv21;
     eq,gv78,1[[gv78,gv76;ml,gv138,gv78,gv21]];
     ad,gv117,gv117,gv11;df,gv118,gv117,gv26;ad,gv119,gv119,gv22;
     eq,gv79,1[[gv79,gv76;ml,gv139,gv79,gv22]];
     eq,gv80,1[[gv80,gv76]];
     df,gv106,gv106,gv26;ad,gv106,gv106,gv11;
     eq,gv81,1[[gv81,gv76]];
     ml,gv71,gv71,12;ml,gv73,gv73,12;ml,gv56,gv56,12;
     ad,gv108,gv108,gv18;ad,gv110,gv110,gv11;
     bx,1,gv122;bx,2,gv122;bx,3,gv122;bx,5,gv128;bx,0,gv122]
```

THIS STYLE GENERATOR will compute and assign values to the general variables which are used in the single-column algorithms and the format statements for chapter heads, running heads, subheads, and secondary text.

The values on which the single-column style generator operates are supplied via a job specification based on the style specification for the title of interest. Two sample job specifications are developed in Chapter Ten.

Elements of the job specification which do not apply for a given job should be simply omitted from the job specification.

```
19 ≡ [df,gv88,gv60,1;ml,gv88,gv88,gv26;ad,gv88,gv88,gv11;
        qo,gv92,gv88,2;
        ml,gv58,gv59,2;ad,gv58,gv58,gv57;
        eq,gv123,1[[gr,gv188,gv15,gv16;ad,gv123,gv188,gv95;
                ad,gv186,gv123,gv88;df,gv121,gv186,gv15;
                df,gv87,gv121,gv94;gv140,gv122]];
        eq,gv123,0[[ad,gv121,gv88,gv94;
                ne,gv72,0[[ml,gv72,gv72,gv15;
                        ml,gv85,gv72,2;
                        ad,gv85,gv85,gv58;
                        ad,gv96,gv72,gv96]];
                ad,gv140,gv122,gv72]];
        ad,gv128,gv140,gv59;ad,gv128,gv128,gv57;ml,pi,gv76,2;
        df,gv173,gv29,gv14;df,gv174,gv14,gv25;ml,gv174,gv174,32;
        ad,gv102,gv102,gv11;df,gv103,gv102,gv26;ml,gv61,gv61,gv12;
        ad,gv104,gv104,gv12;df,gv105,gv104,gv27;
        ad,gv111,gv111,gv11;df,gv112,gv111,gv26;ad,gv113,gv113,gv20;
        eq,gv77,1[[gv77,gv76;ml,gv137,gv77,gv20]];
        ad,gv114,gv114,gv11;df,gv115,gv114,gv26;ad,gv116,gv116,gv21;
        eq,gv78,1[[gv78,gv76;ml,gv138,gv78,gv21]];
        ad,gv117,gv117,gv11;df,gv118,gv117,gv26;ad,gv119,gv119,gv22;
        eq,gv79,1[[gv79,gv76;ml,gv139,gv79,gv22]];
        eq,gv80,1[[gv80,gv76]];
        df,gv106,gv106,gv26;ad,gv106,gv106,gv11;
        eq,gv81,1[[gv81,gv76]];
        ml,gv71,gv71,12;ml,gv73,gv73,12;ml,gv56,gv56,12;
        ad,gv108,gv108,gv18;ad,gv110,gv110,gv11;
        bx,1,gv140;bx,4,gv128;bx,2,gv122;by,2,gv121;
        bx,3,gv140;bx,5,gv128;bx,6,gv140;by,6,0]
```

THIS STYLE GENERATOR will compute and assign values to the general variables which are used in the two-column algorithms and the format statements for chapter heads, running heads, subheads, and secondary text.

The values on which the two-column style generator operates are supplied via a job specification based on the style specification for the title of interest. A sample job specification is developed in Chapter Ten.

Elements of the job specification which do not apply for a given job should be simply omitted from the job specification.

```
G1 ≡ [rm,gv199,pn,2;
    eq,gv199,0[[ps,gv15;tf,gv45;dc,pn;fl;ps,gv16;tf,gv46;z1]];
    eq,gv199,1[[ps,gv16;tf,gv46;z2;fl;ps,gv15;tf,gv45;dc,pn]]]

G2 ≡ [rm,gv199,pn,2;
    eq,gv199,0[[ql;ps,gv15;tf,gv45;dc,pn;
               hn,gv96;ps,gv16;tf,gv46;z1]];
    eq,gv199,1[[qr;ps,gv16;tf,gv46;z2;
               hn,gv96;ps,gv15;tf,gv45;dc,pn]]]

G3 ≡ [qr;rm,gv199,pn,2;eq,gv199,0[[ql]];dc,pn]

G5 ≡ [ce;ps,gv16;tf,gv46;rm,gv199,pn,2;
    eq,gv199,0[[z1;dn,0;ql]];
    eq,gv199,1[[z2;dn,0;qr]];ps,gv15;tf,gv45;dc,pn]

G6 ≡ [ce;dc,pn]

G7 ≡ [rm,gv199,pn,2;

    eq,gv199,0[[lb,0;rb,gv72;ps,gv15;tf,gv45;qr;dc,pn;dn,0;
               lb,gv96;rb,gv85;ps,gv16;tf,gv46;ql;z1]];
    eq,gv199,1[[lb,0;df,rb,gv85,gv96;ps,gv16;tf,gv46;qr;z2;
               dn,0;df,lb,gv85,gv72;rb,gv85;ps,gv15;
               tf,gv45;ql;dc,pn]]]
```

THESE SUBSIDIARY format statements may be used with any of the pagination algorithms to provide the running head treatment desired. (Exception: provision for G7—which provides for folios to hang in the margins—exists only in the two-column style generator.) A given running head is specified for use by equating its label with Y2 in the job specification.

Additionally, the synonym names Z1 and Z2 must be given to the left-hand and right-hand running heads respectively; the appropriate shift levels (eg., for caps, small caps, or cap and lower case) must be included within the synonym definitions.

The sample pages shown on pages B2 and B6 illustrate use of G1 for running heads and G6 for the chapter opening drop folio.

The sample pages shown on page B8 illustrate the use of G2 for running heads and G6 for the drop folio.

The chapter opening pages of the main portion of this book illustrate use of G3 for flush outside drop folio.

The sample pages shown on page B4 illustrate the use of G5 for running heads and G6 for the drop folio.

The sample pages shown on page B10 illustrate use of G7 for running feet.

Note: that whether the result is a running head or a running foot is a function of the pagination algorithm which positions the folio-running head-running foot block; not of the format statements on this page.

```
H1 ≡ [ne,cy,0[[eq,cx,0[[dj,gv112]];ne,cx,0[[dj,gv111]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv111]]]];
      ps,gv20;bl,gv35;tf,gv50;lb,0;ql;hn,gv77]

H2 ≡ [dn,gv113;ps,gv11;bl,gv26;tf,gv41;ju;hn,gv80]

H3 ≡ [ne,cy,0[[eq,cx,0[[dj,gv115]];ne,cx,0[[dj,gv114]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv114]]]];
      ps,gv21;bl,gv36;tf,gv51;lb,0;ql;hn,gv78]

H4 ≡ [dn,gv116;ps,gv11;bl,gv26;tf,gv41;ju;hn,gv80]

H5 ≡ [ne,cy,0[[eq,cx,0[[dj,gv118]];ne,cx,0[[dj,gv117]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv117]]]];
      ps,gv22;bl,gv37;tf,gv52;lb,0;ql;hn,gv79;ob]

H6 ≡ [dn,gv119;ps,gv11;bl,gv26;tf,gv41;ju;hn,gv80;ro]

H7 ≡ [nl;dj,gv106;ps,gv13;bl,gv28;tf,gv43;lb,0;ju;hn,gv81]

H8 ≡ [ps,gv11;bl,gv26;tf,gv41;hn,gv82]
```

THESE SUBSIDIARY format statements may be used with any of the pagination algorithms to provide spacing and font changes for three levels of side heads and a "run in" side head.

If gv170 is assigned a value in the job specification, a sink of this assigned value will precede those side heads which appear first in a text block.

The sample pages shown on pages B2 and B6 illustrate use of H1 and H2 for first level side heads. The pages of the main portion of this book illustrate the use of H3 and H4, H5 and H6, and H7 and H8 for second and third level side heads and run-in side heads, respectively.

```
P1 ≡ [ne,cy,0[[eq,cx,0[[dj,gv112]];ne,cx,0[[dj,gv111]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv111]]]];
      ps,gv20;bl,gv35;tf,gv50;lb,0;ce]

P2 ≡ [dn,gv113;ps,gv11;bl,gv26;tf,gv41;ju;hn,gv80]

P3 ≡ [ne,cy,0[[eq,cx,0[[dj,gv115]];ne,cx,0[[dj,gv114]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv114]]]];
      ps,gv21;bl,gv36;tf,gv51;lb,0;ce]

P4 ≡ [dn,gv116;ps,gv11;bl,gv26;tf,gv41;ju;hn,gv80]

P5 ≡ [ne,cy,0[[eq,cx,0[[dj,gv118]];ne,cx,0[[dj,gv117]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv117]]]];
      ps,gv22;bl,gv37;tf,gv52;lb,0;ce;ob]

P6 ≡ [dn,gv119;ps,gv11;bl,gv26;tf,gv41;ju;hn,gv80;ro]
```

THESE SUBSIDIARY format statements may be used with any of the pagination algorithms to provide spacing and font changes for three levels of center heads.

If gv170 is assigned a value in the job specification, a sink of this assigned value will precede those side heads which appear first in a text block.

The sample pages shown on page B4 illustrate use of P1 and P2 for first level center head.

```
R1 ≡ [ne,cy,0[[eq,cx,0[[dj,gv112]];ne,cx,0[[dj,gv111]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv111]]]];
      ps,gv20;bl,gv35;tf,gv50;lb,0;pw,rb;df,rb,rb,gv137;qr]

R2 ≡ [dn,gv113;ps,gv11;bl,gv26;tf,gv41;rb,pw;ju;hn,gv80]

R3 ≡ [ne,cy,0[[eq,cx,0[[dj,gv115]];ne,cx,0[[dj,gv114]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv114]]]];
      ps,gv21;bl,gv36;tf,gv51;lb,0;pw,rb;df,rb,rb,gv138;qr]

R4 ≡ [dn,gv116;ps,gv11;bl,gv26;tf,gv41;rb,pw;ju;hn,gv80]

R5 ≡ [ne,cy,0[[eq,cx,0[[dj,gv118]];ne,cx,0[[dj,gv117]]]];
      eq,cy,0[[eq,cx,0[[dn,gv170]];ne,cx,0[[dj,gv117]]]];
      ps,gv22;bl,gv37;tf,gv52;lb,0;pw,rb;df,rb,rb,gv139;qr]

R6 ≡ [dn,gv119;ps,gv11;bl,gv26;tf,gv41;rb,pw;ju;hn,gv80]
```

THESE SUBSIDIARY format statements may be used with any of the pagination algorithms to provide spacing and font changes for three levels of side heads.

If gv170 is assigned a value in the job specification, a sink of this assigned value will precede those center heads which appear first in a text block.

D1 ≡ [dn,gv107;ps,gv17;bl,gv32;tf,gv47;lb,gv71;ql]

D2 ≡ [dn,gv108;ps,gv18;bl,gv33;tf,gv48;lb,gv73;ql]

D3 ≡ [dn,gv110;ps,gv11;bl,gv26;tf,gv41;lb,gv56;ju;ar[[lb,0]]]

THESE SUBSIDIARY format statements may be used with any of the pagination algorithms to provide spacing and font changes for two levels of chapter heads and chapter opening.

The sample pages shown on pages B2 and B6 illustrate use of these format statements for a chapter opening.

```
J1 ≡ [ne,cy,0[[eq,cx,0[[dj,gv103]];ne,cx,0[[dj,gv102]]]];
      eq,cy,0[[ne,cx,0[[dj,gv102]]]];ps,gv12;bl,gv27;tf,gv42;lb,gv61]

J2 ≡ [ne,cy,0[[eq,cx,0[[dj,gv105]];ne,cx,0[[dj,gv104]]]];
      eq,cy,0[[ne,cx,0[[dj,gv104]]]];ps,gv11;bl,gv26;tf,gv41;lb,0]

J3 ≡ [ne,cy,0[[eq,cx,0[[dj,gv101]];ne,cx,0[[dj,gv100]]]];
      eq,cy,0[[ne,cx,0[[dj,gv10]]]]]
```

THESE SUBSIDIARY format statements may be used with any of the pagination algorithms to provide spacing and font changes for secondary text.

The sample pages shown on page B6 illustrate use of J1 and J2 for secondary text.

Assignment of general variables

gv1	Column indicator
gv2	Store point size
gv3	Current footnote block identity (two-column text)
gv4	Store typeface
gv5	
gv6	Depth of requested white space (points)
gv7	
gv8	
gv9	
gv10	
gv11	Point size—primary text
gv12	Point size—secondary text
gv13	Point size—run-in side head
gv14	Point size—footnote text
gv15	Point size—folio
gv16	Point size—running head/foot
gv17	Point size—first level chapter head
gv18	Point size—second level chapter head
gv19	Point size—third level chapter head
gv20	Point size—first level sub-head
gv21	Point size—second level sub-head
gv22	Point size—third level sub-head
gv23	
gv24	
gv25	Point size—footnote citation
gv26	Body lead—primary text
gv27	Body lead—secondary text
gv28	Body lead—run-in side head
gv29	Body lead—footnote text
gv30	Point size—drop opening initial
gv31	Typeface—drop opening initial
gv32	Body lead—first level chapter head
gv33	Body lead—second level chapter head
gv34	Body lead—third level chapter head
gv35	Body lead—first level sub-head
gv36	Body lead—second level sub-head
gv37	Body lead—third level sub-head
gv38	
gv39	
gv40	
gv41	Typeface—primary text
gv42	Typeface—secondary text
gv43	Typeface—run-in side head
gv44	Typeface—footnote text
gv45	Typeface—folio
gv46	Typeface—running head/foot
gv47	Typeface—first level chapter head
gv48	Typeface—second level chapter head
gv49	Typeface—third level chapter head
gv50	Typeface—first level sub-head
gv51	Typeface—second level sub-head
gv52	Typeface—third level sub-head
gv53	
gv54	
gv55	Typeface—footnote citation
gv56	Indention—chapter opening line (picas)
gv57	Gutter(s)—multi-column text (points)
gv58	Full two-column measure
gv59	Measure—primary text (points)
gv60	Lines per nominal text column
gv61	Indention—secondary text (ems)
gv62	
gv63	
gv64	
gv65	
gv66	
gv67	
gv68	
gv69	
gv70	
gv71	Indention—first level chapter head (picas)
gv72	Width allotment for hanging folios (ems)
gv73	Indention—second level chapter head (picas)
gv74	
gv75	Indention—third level chapter head (picas)
gv76	Indention—standard paragraph (ems)
gv77	First level side head indented? (1 = yes)
gv78	Second level side head indented? (1 = yes)
gv79	Third level side head indented? (1 = yes)
gv80	Standard paragraph follows sub-heads? (1 = yes)
gv81	Run-in side head indented? (1 = yes)
gv82	Fixed space run-in side head/following text (ems)
gv83	
gv84	
gv85	Folio block measure—hanging folio
gv86	Depth remaining, text—current page
gv87	Depth of text—chapter opening page
gv88	Depth of text—nominal page
gv89	Depth of text block
gv90	Depth—current page (used in computation)
gv91	One-half column depth—current page
gv92	One-half column depth—next page
gv93	
gv94	Space above drop folio (points)
gv95	Space below running head (points)
gv96	Fixed space between folio & running head (ems)
gv97	Fixed space between folio & running head (ems)
gv98	Depth of text—column one (revised)
gv99	Depth of text—column two (revised)
gv100	Space between items in secondary text

gv101	Space between items of secondary text
gv102	Space above secondary text
gv103	Space above secondary text
gv104	Space below secondary text
gv105	Space below secondary text
gv106	
gv107	Space above first level chapter head
gv108	Space above second level chapter head
gv109	Space above third level chapter head
gv110	Space above chapter opening line
gv111	Space above first level sub-head
gv112	Space above first level sub-head
gv113	Space below first level sub-head
gv114	Space above second level sub-head
gv115	Space above second level sub-head
gv116	Space below second level sub-head
gv117	Space above third level sub-head
gv118	Space above third level sub-head
gv119	Space below third level sub-head
gv120	X-coordinate block zero origin
gv121	Y-coordinate block two origin
gv122	X-coordinate block one origin
gv123	Y-coordinate block one origin
gv124	X-coordinate block two origin
gv125	Y-coordinate block two origin
gv126	X-coordinate block three origin
gv127	Y-coordinate block three origin
gv128	X-coordinate block four origin
gv129	Y-coordinate block four origin
gv130	X-coordinate block five origin
gv131	Y-coordinate block five origin
gv132	X-coordinate block six origin
gv133	Y-coordinate block six origin
gv134	
gv135	
gv136	
gv137	Right boundary delta—first sub-head, flush right
gv138	Right boundary delta—second sub-head, flush right
gv139	Right boundary delta—third sub-head, flush right
gv140	
gv141	
gv142	
gv143	Y-coordinate block one origin (revised, current page)
gv144	Y-coordinate block one origin (next page)
gv145	
gv146	
gv147	
gv148	
gv149	Y-coordinate block four origin (revised, current page)
gv150	Y-coordinate block four origin (next page)
gv151	
gv152	
gv153	
gv154	
gv155	
gv156	
gv157	
gv158	
gv159	Measure—two-column footnotes
gv160	Y-coordinate footnote block origin
gv161	Y-coordinate, carryover footnote block origin
gv162	
gv163	
gv164	
gv165	
gv166	
gv167	
gv168	Bottom boundary—first column next page
gv169	Bottom boundary—second column next page
gv170	
gv171	
gv172	Space between primary text & footnotes (points)
gv173	Leading—footnote block
gv174	Computed up, citation in footnote block
gv175	Computed up, citation in text block
gv176	
gv177	
gv178	
gv179	
gv180	
gv181	
gv182	Footnote carryover indicator
gv183	Suppress indicator
gv184	
gv185	Depth of type page (final chapter page)
gv186	Depth of type page (nominal page)
gv187	Depth of type page (chapter opening & nominal page)
gv188	Greater of gv15 and gv16
gv189	Two-column footnote depth (final)
gv190	Footnote line counter (two-column footnotes)
gv191	Two-column footnote bottom boundary
gv192	Depth of footnote
gv193	Depth of footnote carryover to next page
gv194	Depth of footnote carryover from preceding page
gv195	Indicator—white space at bottom column one, current page
gv196	Indicator—white space at bottom column two, current page
gv197	Indicator—white space at bottom column one, next page
gv198	Indicator—white space at bottom column two, next page
gv199	Indicator—even/odd page
gv200	Indicator—white space carried forward to next page

Type specimens

Displayed herein is a representative sample of the fonts used in the preparation of this book. They are displayed in sixteen point.

Video Baskerville (140)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
ﬀﬁﬂﬃﬄ.,:;‘’!?-()*—/–% | []—@_†‡¶§#¡¿

Video Century Expanded (124)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;‘’!?-()*—/–%

Video Century Schoolbook (120)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;‘’!?-()*—/–%

Video Clarendon (370)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;‘’!?-()*—/–%

Video Gael (150)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
ﬀﬁﬂﬃﬄ.,:;‘’!?-()*—/–% | []—@_†‡¶§#¡¿⅛ ¼ ⅜ ½ ⅝ ¾ ⅞
ABCDEFGHIJKLMNOPQRSTUVWXYZ&$1234567890
ÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÄËÏÖÜÇÑáéíóúàèìòùâêîôûäëïöüçñ

Video Gael Italic (151)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890$1234567890
ﬀﬁﬂﬃﬄ.,:;“!?-()*—/-% | []—@_†‡¶§#¡¿⅛ ¼ ⅜ ½ ⅝ ¾ ⅞
ÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÄËÏÖÜÇÑáéíóúàèìòùâêîôûäëïöüçñ

Video Gael Bold (152)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
ﬀﬁﬂﬃﬄ.,:;“!?-()*—/-% ⅛ ¼ ⅜ ½ ⅝ ¾ ⅞
ABCDEFGHIJKLMNOPQRSTUVWXYZ&

Video Gael Bold Italic (153)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
ﬀﬁﬂﬃﬄ.,:;“!?-()*—/-%

Video Garamond (180)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
ﬀﬁﬂﬃﬄ.,:;“!?-()*—/-% | []—@_†‡¶§#¡¿

Video Gothic (130)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;“!?-()*—/-% | []—@_†‡¶§#¡¿⅛ ¼ ⅜ ½ ⅝ ¾ ⅞
ABCDEFGHIJKLMNOPQRSTUVWXYZ&

Video Gothic Bold (131)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;"!?-()*—/–% | []__@_†‡¶§#¡¿⅛ ¼ ⅜ ½ ⅝ ¾ ⅞
ABCDEFGHIJKLMNOPQRSTUVWXYZ&

Video Janson (280)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
ﬀﬁﬂﬃﬄ.,:;‘’!?-()*—/–% | []__@_†‡¶§#¡¿⅛ ¼ ⅜ ½ ⅝ ¾ ⅞
ABCDEFGHIJKLMNOPQRSTUVWXYZ&$1234567890
ÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÄËÏÖÜÇÑáéíóúàèìòùâêîôûäëïöüçñ

Video Janson Italic (281)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
ﬀﬁﬂﬃﬄ.,:;‘’!?-()—/–%*

Video Messenger (522)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;'"!?-()* /¢% | []^@¬+‡<># =

Video Modern (372)

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ﬀﬁﬂﬃﬄ&.,:;‘’!?-()*—/–% | []__@_†‡¶§#¡¿
$1234567890

Video News Gothic (310)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyz$1234567890

.,:;“!?-()*—/–%

Video News Gothic Bold (311)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyz$1234567890

.,:;“!?-()*—/–%

Video Primer (160)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyz$1234567890

ﬀﬁﬂﬃﬄ.,:;“!?-()*—/–%

Video Roma (210)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyz$1234567890

ﬀﬁﬂﬃﬄ.,:;“!?-()*—/–% | []—@_†‡¶§#¡¿

Video Roma Italic (211)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyz$1234567890

ﬀﬁﬂﬃﬄ.,:;“!?-()*—/–% | []—@_†‡¶§#¡¿

Video Roma Semi-Bold (212)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyz$1234567890

ﬀﬁﬂﬃﬄ.,:;“!?-()*—/–%

Video Spectra (340)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;‘‘’’!?-()*—/–%

Video Spectra Bold (341)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;‘‘’’!?-()*—/–%

Video Stymie Medium (260)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;‘‘’’!?-()*—/–%

Video Technica Medium (110)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;‘‘’’!?-()*—/–%

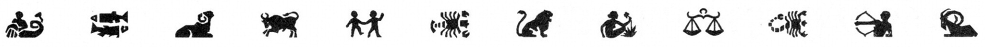
Video Technica Bold (111)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&
abcdefghijklmnopqrstuvwxyz$1234567890
.,:;‘‘’’!?-()*—/–%

Video Utility

⓪①②③④⑤⑥⑦⑧⑨Ⓡ© → ⇄ ← ↓↑ | / √ ¢ °
◆▼◆▲▪■•● ★ ★☆□○ RCA RCA ■ ✠

Video Zodiac Signs



Video Mathematical Greek

ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ —/–

αβγδεζηθϑικλμνξοπρσςτυφϕχψω1234567890

+ − × ÷ = ≡ ≅ ≠ ǂ ~ ≈ ≮ < ⩽ ≶ ≷ ⩾ > ≯

∇ ∞ √ ∴ ′ ″ ‴ ← ∫ → √ ()[]{}

Video Times Roman (100)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&.,:;“!?-()*—/–%

abcdefghijklmnopqrstuvwxyz$1234567890

ﬀﬁﬂﬃﬄ | []__@_†‡¶§#¡¿⅛ ¼ ⅜ ½ ⅝ ¾ ⅞ ^{1234567890} $_{1234567890}$

ABCDEFGHIJKLMNOPQRSTUVWXYZ&$1234567890

ÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÄËÏÖÜÇÑáéíóúàèìòùâêîôûäëïöüçñ

Video Times Roman Italic (101)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&.,:;“!?-()—/–%*

abcdefghijklmnopqrstuvwxyz$1234567890$1234567890

ﬀﬁﬂﬃﬄ | []__@_†‡¶§#¡¿⅛ ¼ ⅜ ½ ⅝ ¾ ⅞

ÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÄËÏÖÜÇÑáéíóúàèìòùâêîôûäëïöüçñ

Video Times Roman Bold (102)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&.,:;“!?-()*—/–%

abcdefghijklmnopqrstuvwxyz$1234567890

ﬀﬁﬂﬃﬄ | []__@_†‡¶§#¡¿⅛ ¼ ⅜ ½ ⅝ ¾ ⅞ ^{1234567890} $_{1234567890}$

ABCDEFGHIJKLMNOPQRSTUVWXYZ&$1234567890

ÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÄËÏÖÜÇÑáéíóúàèìòùâêîôûäëïöüçñ

Video Times Roman Bold Italic (103)

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyz$1234567890$1234567890

ﬀﬁﬂﬃﬄ.,:;‘’!?-()*—/–% | []— @_†‡¶§#¡¿⅛¼⅜½⅝¾⅞

Video Math Times Roman

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyz$1234567890

ﬀﬁﬂﬃﬄ | —@_†‡¶§#⅛¼⅜½⅝¾⅞

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

+ − × ÷ = ≡ ≅ ≠ ǂ ~ ≈ ≮ < ⩽ ⋚ ⋛ ⩾ > ≯

∞ √ ∴ ′ ″ ‴ ← ∫ → √ ()[]{}⟨⟩

Video Display Math

| / ( ) [ ] { } ⟨ ⟩ ∫ √ Σ Π Γ

Video Bridge

♣ ♦ ♥ ♠

A glossary of terms

Two fine reference manuals in the allied fields of typography and electronic data processing are:
A Manual of Style, (11th ed.; Chicago: University of Chicago Press, 1949), and
Automatic Data Processing Glossary, (U. S. Government Printing Office: 1966 O–219–763)

A glossary of terms

A.A., A.C.—Abbreviation of Author's Alterations, Author's Corrections; changes requested by the author after type is correctly set to the original specification.

Accents—Marks over, under, or through particular characters to indicate specific sound value, stress, or pitch, to distinguish words otherwise identically spelled. Standard VideoFont option eight comprises upper and lower case vowels with the accents acute, grave, circumflex, and diaeresis; upper and lower case letter c-cedilla; and upper and lower case n-tilde. Other accented characters are available.

Acute—An accent placed over a letter; for example, ú, to indicate stress.

Agate—A type size, corresponding to 5½ points, chiefly used in classified advertising.

Algorithm—A procedure for solving a problem; it frequently involves repetition of a number of the steps in the procedure.

Alphabet length—The length, in points, of the lower case alphabet of a particular size and style of type. It is used in copy fitting.

Alphanumeric—In general, alphanumeric characters may be alphabetic characters, numerals, or special symbols. In PAGE–1, the alphanumeric set from which format statement and synonym names may be constructed includes only alphabetic characters and numerals.

Antique type—A style of type in which all parts of the letters are of a uniform thickness.

Arabic figures—The characters 1, 2, 3, 4, etc. as distinguished from the Roman I, II, III, IV.

Ascenders—That part of some letters which rises above the central portion of the lower case letters; b, d, f, h, k, and l are ascending letters.

Asterisk (*)—One of the old-style reference marks.

Attributes—In PAGE–1, the term is used to characterize the values of the typographic variables (eg., point size, left boundary, etc.) and the rules of typesetting procedure (eg., justify mode, expand-to-fill-measure, etc.) that are associated with the primary or interrupting block.

BF.—Abbreviation for bold-face type.

Bleed—To "bleed" a page is to cause the printed image to extend to the trim edge(s) of the page.

Blind folio—A page number that is counted but not actually expressed in typesetting the book.

Block—An area on the print region, or page, in which characters and words are set from left to right in a line and line by line downward. It has a location on the print region and "attributes" which state its limits and rules of composition.

Body type—The type used for the main body of a book as opposed to that used for headings.

Cap—An abbreviation of capital.

Caps and small caps—Two sizes of capital letters in one point size; they are sometimes used in combination: Caps and Small Caps. The standard VideoFont option four comprises small caps.

Caption—An explanatory comment, usually above, accompanying a cut or a pictorial illustration.

Cedilla—A mark placed under the letter *c* (as ç) to show that the *c* is pronounced like *s*.

Center head—A sub-head which introduces a section of a book. It is generally set in a different typeface than is text and it is generally provided additional space above and below. It is centered between the margins of the text column.

Character—A single unit of type—a letter of the alphabet, a digit, a punction mark, a space, etc.

Circumflex—The caret-shaped accent mark placed over a letter; for example, â.

Code—1. (*v*) Coding is the part of computer programming in which the program or plan is expressed in the language of the machine or the processor for which the program was designed.
2. (*n*) Sometimes used to mean instruction, or the fixed part of the instruction (ie., without parameters; for example, the code for point size is **ps**).

Command—Sometimes used to mean instruction.

Compose—To assemble, arrange, or otherwise *set* type. Unjustified data, representing the manuscript, are *composed* into lines and from lines, pages are made; both according to the rules stated in the procedures employed.

Condensed—A term used in printing to designate a typeface that is thinner than others in the same family. VideoFonts may be condensed electronically, as is the type in this sentence.

Control code—Sometimes used to mean instruction.

Copy—Hand-written or typewritten words or manuscript from which the typographer works. Also a single specimen of a finished work.

Cut—A term used to denote illustrative matter.

Dagger (†)—One of the old-style reference marks.

Data—Information. In the PAGE–1 context, the terms *raw data* and *input data* denote the material to be composed. Data are usually in machine readable form (eg., on paper or magnetic tape); "data" may or may not include the PAGE–1 instructions which will direct the processor in composing lines and making pages.

Debugging—The process of locating and correcting errors in a procedure or a program. The testing of a program or a procedure with data representative of that with which the program or procedure was designed to operate and making corrections when it fails to meet the specification to which it was designed.

Delimiter—A character which "bounds" or separates. There are two delimiters in PAGE–1, opening and closing, they are represented in manuscript by left and right square brackets respectively. The delimiters separate text from control and are used for nesting within control as well.

Descenders—That part of some letters which extends below the central portion of the lower case letters; g, j, p, q, and y are descending letters.

Diaeresis—Two dots placed over the second of two adjacent vowels to indicate that they are to be pronounced separately; for example, *naïve.*

Double dagger (‡)—One of the old-style reference marks.

Drop folio—A page number placed at the bottom of the page; it may be set centered or flushed to the outside of the page.

Dummy—A sheet, or set of sheets, usually hand sketched at full size, which show in advance the designer's plan for the finished job.

Em—A unit of linear measure equal to the point size of the type in use; that is, a ten point em is 10 points, etc. The term is applied in many other ways as well. For example, em dash, em space, and em square. Historically, sometimes called a *mutton,* or simply a *mut.*

Em align dash—A dash (__) one em long that aligns with the lower edge of the central portion of the lower case letters.

Em dash—A dash (—) one em long that is approximately centered on the x-height.

En—A unit one-half as wide as an em of the same type. Historically, sometimes called a *nut.*

En align dash—A dash (_) one en long that aligns with the lower edge of the central portion of the lower case letters.

En dash—A dash (–) one en long that is approximately centered on the x-height.

Environment—A term that denotes the information saved when a block is interrupted. The term includes the block attributes and such other things as: current horizontal and vertical setting position, all boundary procedures in force, and information with regard to location in the input text stream at the point of interrupt.

Execution—Performance of the set of detailed steps carried out by the processor to produce the result specified by the primitive instruction code.

Expanded, extended—A term used in printing to designate a typeface that is wider than others in the same family. VideoFonts may be expanded electronically, as is the type in this sentence.

Extract—A bit of text that has been copied and is included as substantiation of a point the author wishes to make or for some other reason. It is generally made to stand out in text by use of a smaller point size and, perhaps, by indention as well. Secondary text.

File—An organized collection of information created for a specific purpose.

Flush—Without indention.

Flush paragraph—A paragraph having no indention.

Folio—The page number.

Font—Originally a complete assortment of one size and style of type; usually upper and lower case roman and italic characters, small caps, numerals, punctuation, ligatures, and reference marks. In the VideoComp system, a font comprises digital data from which the machine "draws" characters in a range of point sizes.

Format—The size, form, etc., of a typeset product. The term *format* has come to be an abbreviation of format statement, the name given the PAGE–1 procedure which causes the processor to compose input data in a specific format.

Format statement—A PAGE–1 facility for naming a string of text, or control, or both. The format statement is loaded on the system format file from which it is available to any job run on the system by using its name at the desired point.

Galley—A flat metal tray for holding type after it has been set. A galley proof, or "galley", is a proof taken from type standing in a galley before the type is made up into pages. Single column, nonpaginated, text columns produced by a VideoComp are sometimes called galleys.

Global—Universal. In the PAGE–1 context, format statements in the format file are *global* in that they are available in the system to any job run. Synonyms, on the other hand, are job-oriented and their definitions must be included with the job input at a point which precedes their use.

Grave—An accent mark placed over a vowel to indicate certain sound qualities; for example, ò.

Grotesk type—A letter form, without serifs, which maintains slight variation between thick and thin strokes.

Gutter—The two inner margins of facing pages of a book. The space between columns in a multicolumn format is also called a gutter.

Hanging figures—The old style figures, (1 2 3 4 5 6 7 8 9 0) which do not "line" at their base, are sometimes called hanging figures.

Hanging folio—A page number placed in the margin of the book; usually at the outside margin.

Hanging indention—A form of paragraph which is set with the first line full measure and subsequent lines indented; the hanging indention is used in dictionaries (see Appendix A), catalogs, obituaries, etc.

Indention—The amount of space by which the first line is further from the left margin than are succeeding lines in the paragraph.

Inferior fiigures $(_{23})$—Undersize characters placed at the bottom of the type frame for technical work. Standard VideoFont option seven comprises the inferior digits.

Input—Information, or data, transferred to or being transferred from some external storage medium (as magnetic tape or punched cards) into the internal storage of the computer.

Instruction—A set of characters which, as a unit, causes the processor to perform one of its functions. Some instructions require parameters; in these cases, the primitive code together with the parameter(s) are the instruction.

Italic—The style of type that slopes forward in imitation of cursive writing, as opposed to the upright or roman letter. A pseudoitalic character, called oblique, may be generated by the VideoComp from font data that were digitized roman. of type to be the same length by expanding or condensing the wordspaces in the lines as necessary.

Justification—The process of causing all lines to have a common length. Justification is effected by expansion or contraction of the spaces in the line and additionally, perhaps, by hyphenation.

Justify—To cause succeeding lines of type to be the same length by expanding or condensing the wordspaces in the lines as necessary.

Kerned letter—Letters which have a part that extends over the body of the type, or outside the nominal type frame. For example: the *q* in *Quill.*

Language—A system for representing and communicating information between people or, in the case of computer programming languages, between people and programming systems.

Lead, leading—Strips of metal used to increase the space between lines of type; hence, space between lines of type.

Leaders—Short dashes or dots placed at intervals to guide the eye to figures or words at the ends of open lines. Examples of usage are: tables of contents, indexes, price lists, etc.

LF.—Abbreviation for light-face type.

Ligature—A character consisting of two or more letters tied together. Standard VideoFont option one comprises the five most commonly used ligatures: ff, fi, fl, ffi, and ffl.

Lining figures—Figures that are designed such that they "line" together at their base; as 1 2 3 4 5 6 7 8 9 0. The figures in the standard VideoFont basic character set are lining figures.

Logotype—A single piece of type faced with a special character; for example; a corporate identity symbol. Sometimes called a *logo.* The RCA logo, RCA, actually comprises three distinct, stylized characters.

Macro instruction—A "macro" is an instruction that is replaced in a computer program by a predetermined or generated sequence of machine instructions.

Make-up—Page make-up is the arranging of lines of type, and illustrations, into pages of the specified depth and including the specified running heads and folios.

Measure—The width of a column or page of type; usually specified in picas.

Modern type—A class of roman typeface in which the contrast between thick and thin strokes is greatly increased and in which the serifs are simply thin lines. The best known modern face is Bodoni (by the Italian, Giambattista Bodoni about 1818).

Monospace font—A font of type in which all characters have the same set width.

Nest—To embed a statement or a procedure within a larger procedure. The term, nested conditional procedures, refers to conditional procedures placed within other conditional procedures.

Nest level—The data "level" is called nest level zero and "control" is nest level one. If procedures (or text) are nested within a control string, the nest level changes from one (control), to two (text), to three (control), to four (text), etc. Odd numbered nest levels are control and even numbered nest levels are text.

Old style figures—The old style figures are 1 2 3 4 5 6 7 8 9 0; since they do not "line" at their base, they are sometimes called hanging figures.

Old style type—One of two general classes into which roman typefaces are divided. The other is modern. The best known old style typefaces are Garamond (by the Frenchman, Claude Garamond about 1540) and Caslon (by the Englishman, William Caslon about 1734).

Operator—A mathematical symbol that indicates a mathematical process to be performed. The PAGE–1 primitive codes that represent mathematical operations are called operators.

Output—Information transferred from internal computer storage to an external medium or to a device outside the computer.

Overset—When, in the composition of lines, the accumulated set widths of the characters on the line exceeds column measure. Response to this circumstance is contingent on the justification and hyphenation conventions in force at the time of overset.

Page proof—A proof taken from type which has been made-up into pages.

Pagination—The process of producing final pages from lines of type; it includes the addition of running heads and folios.

Paragraph mark (¶)—One of the old-style reference marks.

Parameter—In a computer program, a parameter is an entity, or a quantity, used to express a value or a condition in a computer program. The parameter may be given a different value, or made to express a different condition, each time the program is executed. Use of parameters in computer program subroutines in general, and in PAGE–1 procedures as well, makes the program or procedure readily adaptable to a variety of uses.

Pica—A unit used by typographers and printers to specify and measure such things as line length and page depth; six picas are approximately equal to one inch.

Point—The unit of type measurement—in the United States, the point is approximately 1/72nd of an inch; twelve points equal one pica.

Point size—The dimension of a type from the top of the ascending letters (b, d, f, h, k, l) to the bottom of the descending letters (g, j, p, q, y).

Primitive code—The lowest level of language code.

Print region—Page.

Procedure—A course of action taken to solve a problem.

Processor—1. A programming system that employs the resources of its computer, according to the instructions of any given program, to solve a problem. 2. The machine itself, or its arithmetic unit.

Proof—A trial printing impression that is used to detect and correct errors.

Program—1. (*v*) To "program" a computer is to devise a plan for solving the problem at hand. *Coding* is sometimes implied in the term, program; but refers to the detailed expression of the program in the language being used. 2. (*n*) A computer "program" is a plan or a routine devised to solve a problem on a computer.

Programmer—A person who prepares problem solving procedures; he may also *code* the program and debug it.

Raised initial—A large initial letter sometimes used at the beginning of sections of printed material. The letters which follow should be "mortised" or "mitered" in order that excessive space not separate the initial capital from the rest of the word. (See Appendix B.)

Recto—Right-hand side, odd numbered pages of a book.

River—Wordspaces, accidentally coinciding in succeeding lines of type, form "rivers" of white in the typeset page.

Roman figures—Numerals expressed by letters as opposed to Arabic numerals: I, II, III, IV, etc.

Routine—A set of computer instructions that carry out some well defined function.

Rules—Thin strips of metal, in varying thickness, used for printing straight lines and borders. Rules are generated in PAGE-1 using the *fill-to-measure* (horizontal) and *execute* (vertical) commands.

Run-around—A blank area in a typeset page in which measures have been adjusted to allow for subsequent insertion of an illustration.

Running foot—The title of a book, a chapter, or major section set at the bottom of each page of a book.

Running head—The title of a book, a chapter, or major section set at the top of each page of a book.

Run-on—A term that describes the combining of one or more sentences to avoid making additional paragraphs.

Sample page—A specimen or an example of a proposed book or other typeset product for purposes of design approval.

Sans serif type—A letter without serifs and constructed generally from strokes of equal thickness.

Section mark (§)—One of the old-style reference marks.

Serif—Any of the short lines that stem from and are at an angle to the upper and lower strokes of a letter.

Set solid—Lines of text material set close together; that is, without additional leading.

Setting position—The position of a given point in a line of words or characters or, vertically, in a column of lines. The horizontal setting position at some point is measured as the accumulated set widths of all the characters set on the line at the point the measurement is taken. The vertical setting position is the depth in the text column to the lower edge of the last complete line set.

Set width—The horizontal dimension of each individual character in a font is known as its *set,* or its set width.

Side head—A sub-head which introduces a section of a book. It is generally set in a different typeface than is text and it is generally provided additional space above and below. It may be set flush left or flush right.

Sink—To provide blank space above or below a typographic entity such as a side or center head, or secondary text, etc.

Small capitals—Capital letters that are approximately the height of the lower case letters in the font. Small capitals are frequently used in running heads.

Space—Blank character. The wordspace is "justifiable" in the composition of justified lines. Em, en, and thin spaces are not altered in line justification.

Straight matter—Plain composition, usually in one size and style of type, as opposed to display matter.

String—A connected sequence of data.

Style specification—A document that governs the typesetting of a book or other document to be set in type.

Sunken initial—A large initial letter "sunken" into an indention in the type lines such that the upper edge of the initial is level with that of the top line of type and the base of the initial appears to "sit" upon the same line as does the last indented line. The treatment given the indented lines varies with the particular sunken initial (See the chapter openings.)

Superior figures $(^{23})$—Undersize characters placed at the top of the type frame for technical work. Standard VideoFont option six comprises the superior digits.

Suspense block—The PAGE–1 system includes one "suspense" block; it has the characteristic that the lines set in it will appear in output with the page that follows the page being made-up when the suspense block is used. It provides the facility to save the carryover portion of a footnote for inclusion on the next page.

Synonym—The PAGE–1 facility for naming a string of text, or controls, or both, with the input data and then specifying its use at a given point by including its *name* at that desired point.

Tabular matter—Composition which involves columns of type; it may also include horizontal and vertical rules.

Text—The body matter of a page or of a book, as distinguished from headings.

Thin space—A fixed space used in the setting of certain types of material; for example, mathematics. The width of a *thin* varies with the typesetting system used; a VideoComp thin space is ¼em in width.

Tilde—A mark placed over the letter *n* (as in the Spanish *señor,* sir) to denote the sound /n^{y}/.

Turnover—A side or center head which will not fit on one line is said to "ture over". The portion of the head which carries to a second line is called a turnover.

Type family—A group of typefaces having the same basic design but varying greatly in weight, thickness, etc. For example: Gael, *Gael Italic,* **Gael Bold, and** ***Gael Bold Italic,*** belong to the same family of type.

Type page—The area on the final typeset page in which characters are set.

Typography—The art of skillful expression through the use of type.

Unjustified—Not justified. Each time a word causes overset, the line is terminated and a new line is started with the word that caused overset. The lines may be set flush left, flush right, or centered.

Variable—A quantity that can assume any of a given set or range of values.

Verso—Left-hand side, even numbered pages of a book.

VideoComp—A family of cathode ray tube phototypesetting machines marketed by RCA Graphic Systems.

White space—Space in which no type is set. Blank space above a side head, for example, is white space. A specified amount of white space may be reserved—and left vacant—for subsequent insertion of illustrative material.

Widow—A single word, in a line by itself, ending a paragraph is a widow. A single line, opening a paragraph, at the bottom of a page or a single line, ending a paragraph, at the top of a page are widows. Widows are frowned upon in quality typography, often difficult to avoid, and present in many fine books.

X–height—The height of the central portion of the lower case letters exclusive of ascenders and descenders.